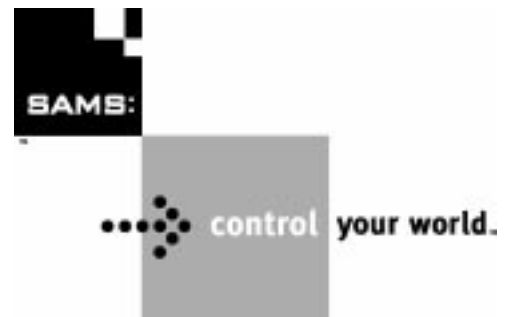




MSP Edition

**Release 9.1
November 1998**

Systems Guide



**© Copyright 1990-1998 Sterling Software, Incorporated
All Rights Reserved**

**Sterling Software, Inc.
Storage Management Division
625 East Carnegie Drive, Suite 100
San Bernardino, California 92408-3510**

SMDS200P004-91

Proprietary/Trade Secret Notice

SAMS:Disk is undergoing constant revision, refinement, and expansion to include additional features. The current documentation is believed to be accurate and reflects the current version of the software. Important information that was not available when this book was published is contained in the Cover Letter that accompanied the distribution tape.

Warranty Disclaimer, Right to Revise

This documentation and the software it describes, constitute the proprietary, confidential, and valuable trade secret information of Sterling Software, Inc., and is licensed either "AS IS" or with a limited warranty, as set forth in the applicable license agreement. NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE.

Sterling Software Inc., reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes. No part of this Message Manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose, without the express written permission of Sterling Software, Inc.

Use, duplication or disclosure by the Government is subject to restrictions as set forth in Sections 52.227-14 and 52.227-19 of the FAR Commercial Computer Software Restricted Rights Clause and/or the Rights in Technical Data and Computer Software Clause at DFAR 252.227.7013 as applicable.

Trademarks

SAMS:Disk[®] MVS Edition is a trademark of Sterling Software, Inc., and has been registered in the United States. Applications in other countries, pending. The SAMS logo and Control Your World are trademarks of Sterling Software, Inc. Applications for registrations are pending in the United States and other countries. Trademarks of other products mentioned in this Systems Guide are held by the companies producing them.

Customer Support

For Users in the United States, contact:

*Sterling Software, Inc.
Storage Management Division
10811 International Drive
Rancho Cordova, CA 95670
(800) 889-0226*

For Users Outside the United States

For technical support please contact your local Sterling Software customer support representative.

Table of Contents

Chapter 1. Files Data Set Management	1
General Information	1
Integrity	2
Physical Blocks	2
Logical Blocks	2
Logical Records	2
Files Data Set Structure	3
Control Blocks	3
Allocation Control Record	3
File Control Block	3
Logical Index Block	3
Logical Data Block	3
Subfiles of the Files Data Set	4
Restrictions	5
Subfile Entry Definition	5
Subfile Definition Parameters	5
Files Data Set Management	7
Formatting the Files Data Set	7
Files Data Set Status Report	7
Field Descriptions — Status Report	8
Files Data Set Capacity	11
Estimating Capacity Requirements	11
Estimating Subfile Capacity Requirements	11
Estimating Physical Space Requirements	11
Performance Considerations	11
Estimating F6425 Physical Space	12
Estimating F6427 Physical Space	14
Capacity Considerations	16
Block Splits	16
Overflow Blocks	16
Extending Allocation	16
Files Data Set Utilities	17
Unload, Reload, And Reorg	17
Unload	17
UNLOAD Command and Parameters	18

RELOAD	19
RELOAD Command and Parameters	20
REORG Subfiles	21
Files Data Set Logging	22
Activation	22
Naming the Log Data Sets	22
Format of the Log Data Sets	23
Managing the Logs	23
Recovery	23
Running the Utility	23
JCL — Files Data Set Recovery Utility	24
FRECOVER Command and Parameters	25
Sample - Files Data Set Recovery Report	26
Subfile Dump Utility	27
Subfile Dump Command for the Files Data Set	27
Sample Subfile Dump Utility	28
Merging Files Data Sets	29
Common File Handler Interface	30
File Handler Interface	31
Function service types	31
Interface for Sequential Reads	33
Processing Support	34
Support Modules and Functions	35
Service Level Supporting Modules	36
Chapter 2. Tape Management	37
Allocation Considerations	38
DDNAMES Used	38
Dynamic Allocation Versus Supplied JCL	38
Producing Duplicate Archive Tapes Concurrently	39
Specifying Blocksizes for Archive/Backup Tapes	40
Creating Multivolume Tape Data Sets	40
EDRC Support in F6470 Cartridge Devices	40
Running Archive/Backup in Simulate Mode	41
Running Multiple Archive/Backup Jobs Concurrently	41
Cataloging the Tape Archives	42
Tape GDG Support (No Longer Recommended)	42
Assigning Tape Expiration Dates	43
Tapepool Considerations	43
Tapepool Updates from PFD	45

Tapepool Update Utility	46
LIST Command	46
ADD Command	47
DELETE Command	47
ERASE Command	47
Tape and Cartridge Switching — Length Considerations	48
For TAPE (F619) Devices	49
For Cartridge (F6470) Devices	49
Tape Format	50

Chapter 3. Tailoring Options 51

General Information	51
Suggested System Parameters	53
System Parameters for Reports:	53
System Parameters for Messages	53
System Parameter for Lines Per Page	54
System Parameter for Separator Pages	54
User Exit for All Sysout Print Lines	54
System Parameter for Files Data Set Logging	54
System Parameters for Archive/Backup	55
System Parameters for Restore/Recover	55
Performance Tips	56
Report Processing	56
PDS Compress	56
Files Data Set	56
I/O Buffering and Memory Requirements	56
Archive/Backup Considerations	58
Activating Data Compression	58
Specifying the Archive Medium	59
Using Multiple Types of Media	60
Naming the Archives	61
JCL Overrides and DSNAME Usage	62
Assigning an Expiration Date to the Archive Data Set	62
Archiving to Disk: Requirements and Recommendations	62
Using Archive Functions to Manage Active Data	67
Archiving the Archives is Not Allowed	68
Archive Integrity and Security	69
Archive Tape Protection	69
Checks Against Overwriting Valid Tapes	69
Tape Status After Abends	70

Archive Disk Protection	71
Processing PDSs that Contain Anomalies	71
Specifying your Options for PDS Error Handling	72
PDS Anomaly Code Descriptions and Notes	74
Processing Defined for Each Anomaly	74
Anomalous Conditions Not Regarded as Errors	81
Error Messages and Associated Processing for Anomalies	81
PFD Custom Reports	82
Coding PFD Report Data Conversion Modules	86
Coding PFD Report Sort Conversion Modules	87
User-Specified Condition Codes	88
DASD Threshold Manager	90
Overview	90
Threshold Manager Control Statement Syntax	91
Threshold Manager Control Hierarchy	94
JCL - Threshold Manager	95
Implementation Considerations	99
Threshold Manager User Exit	100

Chapter 4. Sysparms 103

General Information	103
System Parameter Names Reserved for Users	104
Overriding Sysparms Instream	104
System Parameters	105
Sysparm Definitions	123

Chapter 5. User Exits 213

General Information	213
Using Previous Releases of User Exits	214
User Exits Cross Reference Matrix	215
User Exit Descriptions	221

Chapter 6. Message Formatting 315

Module Name — ADSUT028	315
Message Formatting	316

Chapter 7. Problem Analysis/Aids	319
When You Call In	319
Sending Information to the Technical Support Center	323
Diagnostic Tools	324
Module Intercept Control Stack	324
Finding the Module Intercept Control Stack	324
Example of Module Intercept Control Stack	325
MI Fast Path	326
DMS Diagnostic Area	328
Module Intercept Tracing	328
MI Trace Control Statement Format	329
MI Trace Output	332
Archive Tape Analysis Utility	333
Executing the Utility	333
Modifying the SYSUT1 DD Statement	334
Command Input	334
Sample Report	335
User Exit — Archive Analysis	336
Archive Extract Utility	336
Output from the Extract Utility	337
Executing the Extract Utility	338
EXTRACT command	338
JCL Example	338
Modifying the SYSUT1 and SYSUT2 DD Statements	339
Archive Fix Utility	339
Modifying the SYSUT1 DD Statement	340
Command Input	341
User Exit — Fix Utility: Parameters Passed	342
Repair Archvols (REPARCH) Utility	343
Processing Considerations	343
Output	343
JCL — REPARCH Utility	343
REPARCH Command and Parameters	344
Load Library Status Utility	346
Loadlib Command and Parameters	346
Track Dump Utility	347
TRACKDMP Command and Parameters	348
Diagnostic DD Statements	349
Archive Tape Recovery Utility	350

Chapter 8. Modifying SAMS:Disk	353
Assembly and Link-Edit of SAMS:Disk Modules	353
Issuing Messages	355
Relinking TSS and PFD Control Modules	355
Adding Non-SAMS:Disk Modules	355
Coding for an Interactive Environment	355
Module Call Interface	356
Parameter Library Support - Common Parmlib Interface	357
Common Sysparms Interface	359
System Parameter Validation - Parmedit Member	360
Adding User Functions to SAMS:Disk PFD	362
Modifying SAMS:Disk PFD Functions	364
 Chapter 9. DSCL Processing	 365
Control Block Structure	366
 Chapter 10. Security Processing	 369
Statement of Security Direction	369
Security Overview	369
Resources Protected by the SAMS:Disk Base System	372
Resources Protected by the Security Interfaces	373
Resources not Protected by SAMS:Disk	375
Security and the SAMS:Disk Auto-Restore Function	376
SAMS:Disk Security Interface General Information	377
Running Without a SAMS:Disk Security Interface	378
Password-Indicated Data Set Support	378
What Authority Does SAMS:Disk Look For?	378
Authority Level Translation Tables for RACF	379
Protection of Commands Under RACF	381
Protection of User Data Sets Under RACF	384
Functions that Need no Data Set/Volume Authorization	393
RACF Discrete Profile Handling	393
RACF Terms Defined	393
RACF and VSAM Data Sets	394
Backing Up RACF Discrete Profiles (RACFBKUP)	394
Restoring RACF-Indicated Data Sets Having no Profiles	395
Reconciling SAMS:Disk and RACF Records	396
Check Command Parameter Description	397
RACF Checking Logic Summary in SAMS:Disk	399

Use of RACF Macros	400
RACF User Exits	400
RACF Installation Exits	401
DSNINDEX RACF Flags	402
Format of SAMS:Disk-Saved Profiles	402
RACF Encode Profile (Format 1)	403
RACF Encode Profile (Format 2)	403
Conversion of SAMS:Disk RACF Profile Encoding	404
Management of SAMS:Disk-Saved Profiles	404
SAMS:Disk-Saved Profile List Utility 1	405
DD Statement Explanations:	406
Select Command	406
SAMS:Disk-Saved Profile List Utility 2	407
DD Statement Explanations:	408
Select Command	408
RACF Profile to Archive Dsname X-REF Reports	409
RACF Profile to Archive Dsname X-REF Reports	409
SAMS:Disk-Saved Profile Maintenance Utility	410
Display Command	411
Delete Command	412
Create Command	412
Rename Command	413
Execution Examples:	414
Changing the Value of RACFDVOL	415

Chapter 11. Report Processing 417

REPORTNAME Table — Member REPORTNM	417
Format-1 / Format-4 DSCB Appendage Fields	418
Tables for Report Formatting	419
Format Table Examples	423
Header Entry Examples	423
Detail Entry Examples	424
Total Entry Examples	425
Report Modification Example	426
Example of Owner Table	427
Standard Multiple Volume Dictionary Report Table	428
Example of Record Format for Owner Record	429
Standard Multiple Volume Dictionary Record Format	430
Report Logic Diagram	432
Report Processing - ISAM	435

ISAM Space Restrictions	436
VTOC ENTRIES- Format 1, 2 and 3 DSCBs	437
ISAM Report Logic Overview	438
ISAM Variables Examined	440
ISAM Format-2 DSCB Offsets	442

Chapter 12. Algorithms 445

Bytes that Fit on an Output Track	446
VSAM Idle Space Calculation	446
VSAM New Name Algorithm	447
Data Compression/Decompression	449
Data Compression Techniques	451
Data Compression Technique 0 (DCCMPEX0)	452
Compression Results	453
Overview of Space Allocation for Archival to DISK	454
Space Allocation Algorithm	454
General Comparison of Allocation in Prior Releases	455
Creating a Disk Archive Data Set	457
PDS Compression Threshold Calculations	459
Directory Used Space Percentage	459
Member Data Used Space Percentage	459
PDS Compression Space Calculations	460
Directory Block Allocation	460
Member Data Space Allocation	461
Secondary Space Allocation	462

Chapter 13. Record Formats 463

Report Record Format - (RPTNALLOCS)	463
Volume Allocation Summary Report — ALLOCS	463
Table Used for Printing ALLOCS Report	464
Report Record Format — RPTNAMVDIC	466
A-FORM of MVDICT Report	466
Table Used to Print MVD Report by Owner	467
Report Record Format — (RPTNATTRBS)	469
Volume Attribute Summary Report	469
Table Used to Print ATTRIBS Report	470
Report Record Format — (RPTNBMVDIC)	472
B-FORM MVDICT Report	472
Table Used to Print MVDICT Report	473

Report Record Format — (RPTNDEVDSB)	475
Billing Device Report	475
Table Used for Printing Billing Device Report	476
Report Record Format — (RPTNDISTR)	477
Volume Distribution Summary Report	477
Table Used for Printing DISTR Report	478
Report Record Format — (RPTNDSNDSB)	480
Billing by User Code	480
Table Used for Printing Billing by User Code	481
Report Record Layout — (RPTNDSNX)	483
Record Report Format	484
Report Record Layout — (RPTNVOLX)	485
Report Record Format	486
Report Record Format — (RPTNDSINDEX)	487
Data Set Index Report	487
Table Used for Printing Data Set Index Report	487
Report Record Format — (RPTNDSUTIL)	488
Data Set Utilization Report	488
Table Used for Printing DSUTIL Report	488
Report Record Format — RPTNDAYWEK	490
Utilization by Date and Time Report	490
Table Used for Printing DAYWEK Report	490
Report Record Format — (RPTNDFREE)	492
Volume Freespace Report	492
Table Used to Print Volume Freespace Report	493
Report Record Format — (RPTNLAYOUT)	494
Volume Layout Report	494
Table Used for Volume Layout Report	495
Report Record Format — (RPTNMAPP)	496
Volume Mapping Report	496
Table Used to Print Volume Mapping Report	497
Report Record Format — (RPTNMEMBER)	498
PDS Directory / Member Report	498
Table Used to Print MEMBER Report	499
Report Record Format — (RPTNMVDICT)	500
Multiple Volume Data Set Dictionary Report	500
Table Used to Print MVDICT Report	500
Report Record Format — (RPTNPDSTAT)	502
Partitioned Data Set Status Report	502
Table Used to Print PDS Status Report	502

Report Record Format — (RPTNPOCCHH, RPTNPODUMP, and RPTNPOTTRD)	504
Partitioned Data Set Directory (CCHHR, DUMP, TTRD)	504
Report Record Format — (RPTNPOXREF, RPTNPO1REF, and RPTNPO2REF)	505
Data Set Name Cross-Reference for PDS Member Names	505
Report Record Format — (RPTNRACFDN)	505
RACF Profile X-REF in Dsname Sequence	505
Table Used to Print X-REF Report	506
Report Record Format — (RPTNRACFPF)	507
RACF Profile X-REF in Profile Name Sequence	507
Table Used to Print X-REF Report	508
Report Record Format — (RPTNSVDICT)	509
Single Volume Dictionary Report	509
Table Used to Print SVDICT Report	509
Report Record Format — (RPTNVOLDSB)	511
Billing Volume Report	511
Table Used for Printing Billing Volume Report	512
Report Record Format — (ARCHDSNVOL)	513
Archive / Retain Report	513
Table Used to Print Archive Report	514
Report Record Format — (ARCHVOLDSN)	515
Archive / Retain Report	515
Table Used to Print Archive / Retain Report	516
Report Record Format — (MIGRDSNVOL)	517
Migrate Report	517
Table Used for Printing Migrate Report	518
Report Record Format — (MIGRVOLDSN)	519
Migrate Report	519
Table Used for Printing Migrate Report in VOL/DSN Order	520
Report Record Format — (VSAMCLDATA)	521
VSAM Cluster Data Record	521
Report Record Format — (RPTDSNEX)	523
VBACKUP Extent Report	523
Table Used for Printing VBACKUP Extent Report	524
Subfile Record Format — (ARCHCMDS)	525
Deferred Archive Requests	525
Subfile Record Format — (ARCHVOLS)	526
Subfile Record Format — (DASDSPCB)	527
DASD BILLING	527

Subfile Record Format — (DMSPARMS)	528
TSS Dynamic Restore	528
Subfile Record Format — (DMSPOLS)	528
For Tapes in the Tape Pools	528
Subfile Record Format — (DSNINDEX)	529
For Archived Data Sets	529
Subfile Record Format — (MIGRECAT)	532
PS Migration to Tape	532
Subfile Record Format — (RACFENC)	532
RACF Encoded Profile Names	532
Subfile Record Format — (RESTCMDS)	533
Deferred Restore Requests	533
Subfile Record Format — (RETEXCLD)	534
Grace Periods for Restored Data Sets	534
RECORD FORMATS — FILES DATA SET LOG RECORDS	534
FILES DATA SET LOG RECORD LAYOUT	534
Archive Tape Organization	535
Archive Tape Format	535
Archive Tape Format (Volume-level Processing)	541
Archive Tape Format for Volume-level Backup Tapes	542

Chapter 14. PARMLIB 543

Purpose	543
Updating Member Data	544
Backup Considerations	544
PARMLIB Specifications	545
Special Rules for NAMEREFS	546
Special Rules for POOLDEFS	547
PARMLIB Syntax Validation Utility	547
User PARMLIB Members	548
Billing Members in PARMLIB	549
Billing User Code Definition Member	549
DASD Pool List	550
Auto-Restore DSNAME Pool Member	551
Auto-Restore Source Volume Pool Member	552
Exclusion Table for Data Compression	553
Inclusion Table for Data Compression	554
Exemption Lists - Generalized	554
Exemption List for Implicit Archive (Retain)	555
GDG DASD Generations List	556

Migration Exemption List	557
PDS Anomaly Table	558
Reblocking Member	558
Retention Period By Prefix List	558
Retention Control Volume Specification Table	559
Security List for SAMS:Disk Data Set Protection	560
Selection Lists — Generalized	561
System Parameter List	562
TSS/PFD USERID Authorization List	563
User-Specified Condition Codes	564
 Chapter 15. JCL DD Statements	 567
DD Statement Explanations	567
 Appendix A. DSCB Appendage Fields	 573
Format-1 Fields	573
Format-4 Fields	575
 Appendix B. Contents of INSTALL Library	 577
 Appendix C. ENQ/DEQ/Reserve Usage	 583
 Index	 585

List of Figures

Figure 1-1. JCL to Create/Format the Files Data Set	7
Figure 1-2. JCL for Files Data Set Status Report	7
Figure 1-3. Sample Files Data Set Status Report	8
Figure 1-4. Unload Files JCL	18
Figure 1-5. JCL to RELOAD the Files Data Set	19
Figure 1-6. Files Data Set Logging DSNNAME	22
Figure 1-7. Files Data Set Recovery Utility JCL	24
Figure 1-8. Sample Files Data Set Recovery Report	26
Figure 1-9. Subfile Dump Utility JCL	27
Figure 1-10. Sample Subfile Dump Utility	28
Figure 1-11. Call to Retrieve Data Set Index Record	30
Figure 2-1. Sample JCL to Execute Tapepool Utility	46
Figure 2-2. Predicting Output Device Capacities	48
Figure 3-1. Generated 22 Character DSNNAME	61
Figure 3-2. PDS Anomaly Table	73
Figure 3-3. Threshold Manager JCL	95
Figure 4-1. JCL to Override SYSPARMS	104
Table 4-1. Possible Values for ARBACKUP	125
Figure 4-2. List of Reports Controlled by RPTLINES	181
Figure 5-1. Code to Stump User Exit Functions	214
Figure 7-1. Module Intercept Control Stack Listing	325
Figure 7-2. JCL to Activate Module Intercept Tracing	329
Figure 7-3. JCL to Execute Archive Tape Analysis	333
Figure 7-4. Sample Archive Tape Verification Report	335
Figure 7-5. JCL to Execute the Extract Utility	338
Figure 7-6. JCL to Execute Archive Fix Utility	340
Figure 7-7. JCL to Execute the REPARCH Utility	343
Figure 7-8. JCL to Execute Loadlib Status Utility	346
Figure 7-9. JCL to Execute Trackdmp Utility	347
Figure 7-10. JCL to Execute Tape Recovery Utility	350
Figure 8-1. Assembly JCL: Member ASMDISK	354
Figure 8-2. Link JCL: Member R900LINK	354
Figure 8-3. Example of Call Interface	356
Figure 8-4. Common Parmlib Interface Module Example	357
Figure 8-5. Common Interface DM175 Example	359
Figure 8-6. Example of SPFMENU Member of Parmlib	362
Figure 8-7. Sample Dynamic Menu Builder	363
Figure 9-1. Control Block Sample Structure	367

Figure 9-2. Sample Input Command Structure	367
Figure 10-1. Multi-level approach to security.	371
Figure 10-2. RACF Checking Logic Summary Flowchart	399
Figure 10-3. JCL to Convert SAMS:Disk RACF Profiles	404
Figure 10-4. JCL to Execute Profile List Utility 1	406
Figure 10-5. Example RACF Profile List Utility 1	407
Figure 10-6. Example JCL to Execute Profile Utility 2	408
Figure 10-7. Example Execution of the RACF Utility 2	409
Figure 10-8. Sample Profile X-REF Report	409
Figure 10-9. Sample X-REF Report by Profile Name	410
Figure 11-1. Report Logic Diagram	432
Figure 11-2. Record Collection portion of Report Logic	433
Figure 11-3. Detail Records Grouped Together Example	434
Figure 12-1. General flow of data compression	449
Figure 12-2. Data Compression Technique 0 Flowchart	452
Figure 14-1. JCL to Backup PARMLIB	544
Figure 14-2. JCL to Restore PARMLIB from Backup Tape	544
Figure 14-3. Sample JCL to Execute Syntax Validation	547
Figure C-1. Enqueue/Dequeue Reserve Usage Flowchart	583
Figure C-2. Enqueue/Dequeue for VBACKUP/VRECOVER	584

List of Tables

Table 1-1. FILEDEFN Record Format	5
Table 1-2. Files Data Set Physical Space on a F6425	12
Table 1-3. F6425 Device Characteristics	12
Table 1-3. F6425 Device Characteristics	13
Table 1-4. Files Data Set Physical Space on a F6427	14
Table 1-5. F6427 Device Characteristics	15
Table 3-1. Message Suppression Sysparms	53
Table 3-2. DSUTIL Report Sorting Options	53
Table 3-3. Files Data Set Logging Activation Sysparms	54
Table 3-4. Common Archive/Backup Sysparms	55
Table 3-5. Common Restore/Recover Sysparms	55
Table 3-6. Data Compression Sysparms	58
Table 3-7. Output Media Sysparms	60
Table 3-8. Available Values for Media Option Sysparms	60
Table 3-9. PDS Anomaly Codes and Descriptions	74
Table 3-10. PFD Custom Report Field Names	82
Table 3-11. Parameters Passed to Dictionary Module	86
Table 3-12. Parameter Passed to Module for Sorting	87
Table 3-13. Format of \$FLGXyz	89
Table 3-14. Format of MMMMBsc	89
Table 3-15. Valid entries for MMMMBsc	90
Table 3-16. Threshold Manager Syntax Error Messages	92
Table 3-17. Device Type Character Code Values	99
Table 4-1. Possible Values Available for ARCHSORT	127
Table 4-2. Possible Values Available for ARCTNAME	128
Table 4-3. List of Possible Value for DYNCUNIT	146
Table 4-4. List of Possible Values for DYNTUNIT	147
Table 4-5. List of Possible Value for DYNnUNIT	148
Table 4-6. Possible Values for sysparm MIGRSORT	159
Table 4-7. Possible Values Available for RACFALLO	165
Table 4-8. Possible Results when using RESRCAT	175
Table 4-9. SECURDRS Authorization Levels	182
Table 4-10. Possible Values Available for SYSPARMS	189
Table 4-11. Vtoc Reader Names and Descriptions	211
Table 5-1. Parameters Passed to ALLOEXIT	221
Table 5-2. Parameters Passed to ALOCATEX	222
Table 5-3. Parameters Passed to ALOPOSEX	223
Table 5-4. Parameters Passed to ARCDNEX	224

Table 5-5. Parameters Passed to ARCEODEX	225
Table 5-6. Parameters Passed to ARCHIVEX	226
Table 5-7. Parameters Passed to ARESPREX	227
Table 5-8. Parameters Passed to ARPOOLEX	228
Table 5-9. Parameters Passed to DARCHIEX	229
Table 5-10. Parameters Passed to DCCMPEXn	230
Table 5-11. Parameters Passed to DCDCPEXn	230
Table 5-12. Parameters Passed to DCDSNDEX	231
Table 5-13. Parameters Passed to DEFARCEX	232
Table 5-14. Parameters Passed to DERASEEX	233
Table 5-15. Parameters Passed to DINXUFEX	234
Table 5-16. Parameters Passed to DRESTOEX	235
Table 5-17. Parameters Passed to DSBUSREX	236
Table 5-18. Parameters Passed to DSCBCVEX	237
Table 5-19. Parameters Passed to DSCLSCEX	238
Table 5-20. Parameters Passed to DSDISPEX	240
Table 5-21. Parameters Passed to DSNDELEX	241
Table 5-22. Parameters Passed to DSNDLPEX	242
Table 5-23. Parameters Passed to FMSSAFEX	243
Table 5-24. Parameters Passed to FMSSPREX	243
Table 5-25. Parameters Passed to GDGCATEX	244
Table 5-26. Parameters Passed to ICHEKAEX	245
Table 5-27. Parameters Passed to ICHEKPEX	246
Table 5-28. Parameters Passed to IXCTLGEX	247
Table 5-29. Parameters Passed to IXMREVEX	248
Table 5-30. Parameters Passed to LISTDMEX	249
Table 5-31. Parameters Passed to LISTDVEX	250
Table 5-32. Parameters Passed to LISTREEX	251
Table 5-33. Parameters Passed to MERDSNEX	251
Table 5-34. Parameters Passed to MERVOLEX	252
Table 5-35. Parameters Passed to MIGDISEX	253
Table 5-36. Parameters Passed to MIGGDGEX	254
Table 5-37. Parameters Passed to MIGREXIT	254
Table 5-38. Parameters Passed to RAADDAEX	255
Table 5-39. Parameters Passed to RAADDPEX	256
Table 5-40. Parameters Passed to RACFDAEX	257
Table 5-41. Parameters Passed to RACFDPEX	258
Table 5-42. Parameters Passed to RACFEAEX	259
Table 5-43. Parameters Passed to RACFEPEX	260
Table 5-44. Parameters Passed to RACFSMEX	261

Table 5-45. Parameters Passed to RACHKAEX	262
Table 5-46. Parameters Passed to RACHKPEX	263
Table 5-47. Parameters Passed to RADEFAX	264
Table 5-48. Parameters Passed to RADEFPEX	265
Table 5-49. Parameters Passed to RADELAEX	266
Table 5-50. Parameters Passed to RADELPEX	267
Table 5-51. Parameters Passed to RECPRIEX	268
Table 5-52. Parameters Passed to RESAFTEX	269
Table 5-53. Parameters Passed to RESPRIEX	270
Table 5-54. Parameters Passed to RETAFTEX	271
Table 5-55. Parameters Passed to RETGDGEX	272
Table 5-56. Parameters Passed to RETPRIEX	273
Table 5-57. Parameters Passed to RLSEAFEX	274
Table 5-58. Parameters Passed to RPTDSBEX	275
Table 5-59. Parameters Passed to RPTDSNEX	275
Table 5-60. Parameters Passed to RPTDSUEX	276
Table 5-61. Parameters Passed to RPTMVDX	277
Table 5-62. Parameters Passed to SCRTCHX	278
Table 5-63. Parameters Passed to SECURPEX	279
Table 5-64. Parameters 5 - 8 (DSNAME) for SECURPEX	281
Table 5-65. Resource Type = 3 (VOLSER) for SECURPEX	281
Table 5-66. Resource Type = 4 (VSAM Catalog Name)	282
Table 5-67. Resource Type = 5 (Commands) for SECURPEX . . .	282
Table 5-68. Resource Type = 6 (VSAM Component)	283
Table 5-69. Resource Type = 7 (Subfile) for SECURPEX	283
Table 5-70. Parameters Passed to SPACECEX	284
Table 5-71. Parameters Passed to SPFHDGEX	285
Table 5-72. Parameters Passed to SPFORDEX	285
Table 5-73. Parameters Passed to SPFORSEX	286
Table 5-74. Parameters Passed to SYSOUTEX	287
Table 5-75. Parameters Passed to TAOPPOEX	288
Table 5-76. Parameters Passed to TAOPPREX	289
Table 5-77. Parameters Passed to TAPEDLEX	290
Table 5-78. Parameters Passed to TMSCTLEX	291
Table 5-79. Parameters Passed to TPMGMTEX	295
Table 5-80. Parameters Passed to UNLOADX	296
Table 5-81. Parameters Passed to USERSENQ	297
Table 5-82. Parameters Passed to USERVENQ	298
Table 5-83. Parameters Passed to USRCODEX	299
Table 5-84. Parameters Passed to VCCOPYEX	300

Table 5-85. Parameters Passed to VCDMIGEX	301
Table 5-86. Parameters Passed to VSACESEX	303
Table 5-87. Parameters Passed to VSALLOEX	304
Table 5-88. Parameters Passed to VSATTREX	305
Table 5-89. Parameters Passed to VSMAFTEX	307
Table 5-90. Parameters Passed to VSMPRIEX	308
Table 5-91. Parameters Passed to VSMSPWEX	310
Table 5-92. Parameters Passed to VSNEWNEX	311
Table 5-93. Parameters Passed to XCPARCEX	312
Table 5-94. Parameters Passed to XCPDSNEX	313
Table 6-1. Parameters Available to Module ADSUT028	315
Table 6-2. Format of the DMSMSGES Member of PARMLIB	316
Table 7-1. Module Intercept Control Stack Fields	325
Table 7-2. MI Trace Facility DD Statements	328
Table 7-3. Format of the MI Trace Control Statement	329
Table 7-4. MI Trace Options and Definitions	331
Table 7-5. MI Trace Output Explanation	332
Table 7-6. Archive Analysis User Exit	336
Table 7-7. Archive Fix Utility User Exit	342
Table 7-8. REPARCH Command Parameter Descriptions	344
Table 7-9. Archive Tape Recovery Condition Codes	351
Table 8-1. Description of the MI CALL Structure	357
Table 8-2. Field Descriptions within DM175	359
Table 8-3. Format of PARMEDIT Entries	360
Table 10-1. List of RACF User Exits	401
Table 11-1. REPORTNM Content Descriptions	417
Table 11-2. REPORTNM Example	418
Table 11-3. Control Entry Description	419
Table 11-4. Header Entry Description	419
Table 11-5. Detail Entry Description	420
Table 11-6. Possible Line Type Values and Descriptions	421
Table 11-7. Total Entry Descriptions	421
Table 11-8. Special Field Types for Position 4	422
Table 11-9. RPTNOMVD Example	423
Table 11-10. Header Entry Example	423
Table 11-11. Detail Entry Example #1	424
Table 11-12. Detail Entry Example #2	424
Table 11-13. Total Entry Example #1	425
Table 11-14. Total Entry Example #2	425
Table 11-15. Parameters Passed to module ADSDM033	426

Table 11-16. Record Format for the Owner Record	429
Table 11-17. Record Format of the Standard MVD	430
Table 11-18. Four Major Areas of ISAM data sets	440
Table 11-19. ISAM Variable Combinations	441
Table 12-1. Data Compression Techniques available	451
Table 13-1. List of Valid Logical Record Types	536
Table 13-2. Record Types Generated for PS and DA	537
Table 13-3. Record Types Generated for PDS and PDSE	538
Table 13-4. Record Types Generated for ISAM	538
Table 13-5. Record Types Generated for PO	538
Table 13-6. Record Types Generated for VSAM	539
Table 13-7. Record Format of Logical Record Types	539
Table 13-8. Archive Tape Format (Volume-level)	541
Table 13-9. Tape Format for Volume-level Backups	542
Table 15-1. Symbolic Parameters and Recommendations	567
Table A-1. Format-1 DSCB Appendage Fields	573
Table A-2. Format-4 DSCB Appendage Fields	575
Table B-1. List of PROCS in SAMS:Disk Proclib	577
Table B-2. List of PARMLIB Members and Descriptions	579

Chapter 1. Files Data Set Management

Commands	Program Executed
INITIALIZE (assumed)	ADSDM100
STATUS (assumed)	ADSDM107
UNLOAD	ADSDM177
FRECOVER	ADSDM650
RELOAD	ADSDM192
for RESTART	ADSDM192
for REORG	ADSDM444

The following procedures are available:

- UNLOAD
- RELOAD
- REORG
- RESTART

General Information

SAMS:Disk uses a direct access data set called the Files Data Set to store information about the data sets it operates on. It is a single volume data set that must be in one extent. The files data set contains multiple subfiles of data. The term "subfile" refers to a group of logically related records kept within the files data set. Each subfile is assigned an eight-character name by which it is identified. Subfiles are independent groups of records that do not necessarily have any relationship to records in other subfiles of the data set. They are processed as if they were separate data sets. This technique provides the convenience of having multiple DASD files without the task of having to support each one independently.

An EXCP I/O interface is used to retrieve, insert, update and delete records. Records are referenced by their specific keys within each subfile, but may also be retrieved sequentially by subfile name alone, if desired.

For example, one subfile used by SAMS:Disk is the data set index record subfile (DSNINDEX). This subfile contains one record for each data set currently in the SAMS:Disk archives. When a data set is archived, a record is placed in the DSNINDEX subfile. When the data set is restored, the DSNINDEX subfile is accessed to determine the volume serial of the tape or disk containing the archived version of the data set and its location within that volume. These records remain in the subfile until they are deleted by the MERGE or Index maintenance processing routines.

Integrity

The processing technique implemented ensures data set integrity during concurrent update (see "Basic Conventions and Requirements") in both the Loosely Coupled Multi Processor (LCMP) and Tightly Coupled Multi Processor (TCMP) environments. For greatest efficiency, a cross system enqueue package is recommended in a JES environment.

For related information, please turn to the sysparm description for *RSUPPRES* on page [181](#) in this manual.

Except for specified utility functions, all JCL DD statements that reference the files data set may have shared disposition (DISP=SHR) coded in the JCL statement.

Physical Blocks

The physical blocksize of the files data set is the only value common to all of its subfiles. The blocksize is assigned at files initialization time.

Logical Blocks

Internal to SAMS:Disk processing is the concept of Logical Blocks. A logical block is simply a set of contiguous physical blocks accessed by the EXCP interface. The files data set EXCP I/O interface automatically determines the Logical Block based on the capacity requirements used when the files data set is initialized.

The Logical Block is unique to each subfile unlike the physical blocksize which is common to all subfiles. Depending on the capacity requirements for each subfile, the number of physical blocks per Logical Block may vary. The DSNINDEX subfile currently utilizes the maximum Logical Block which is 10 physical blocks per Logical Block.

Each logical block is accessed by Relative Byte Address (RBA). The relative byte address of each logical block is stored in the first four bytes of each block.

Logical Records

This term describes the data records that are stored in the logical data block. One logical record represents one subfile record. The logical data block will contain a fixed number of logical records.

Files Data Set Structure

Control Blocks

The files data set contains 4 major control blocks. The Allocation Control Record, the File Control Block, the Logical Index Block and the Logical Data Block. At file initialization, each subfile is represented by a subfile Logical Index Block and a subfile Logical Data Block. (The files data set control blocks and their contents are further described in the Record Format section, beginning on page [463](#) in this manual.)

Allocation Control Record

The Allocation Control Record (ACR) describes the allocation of the entire files data set. It always resides in the first physical block of the files data set and is predefined offsets into the block.

File Control Block

The File Control Block (FCB) is defined for each subfile and contains the subfile definition entry. (Used by the Files Status Report.) The FCB is divided into two parts. A FCB prefix defines the number of subfiles defined and always follows the allocation control record.

Logical Index Block

Only one Logical Index Block (LIB) is created for every subfile defined by the FILEDEFN member. The LIB contains one entry for every logical data block in the subfile. The logical data block entry contains the highest key value and a four byte RBA pointer to the logical data block that contains the key.

To retrieve a specific record, the logical index block is searched to determine the relative byte address of the logical data block that has a high key equal to or greater than the key of the desired record. The RBA is then used to read the logical data block into memory. The logical data block is then searched for the record.

Logical Data Block

The Logical Data Block (LLB) contains logical records comprising the subfile. The number of logical records that reside in a single logical data block is equal to the blocking factor of the subfile.

Note: A logical data block will never contain a partial record.

Subfiles of the Files Data Set

ARCHCMDS

A record is placed in this file for each ARCHIVE command to be processed on a deferred basis; this includes TSS and batch requests. The deferred command requests are eventually deleted subsequent to a batch DISPOSE job that processes the queued requests.

ARCHVOLS

A record exists in this file for each archive volume (tape or disk) to which one or more data sets were archived. An entry is also present for each backup copy of the archive volumes.

DASDSPCB

The accumulation of DASD space billing statistics are kept in this file.

DMSPARMS

This file is used only if the TSS DYNAMIC (immediate) RESTORE function is used. It is used to control the number of concurrent dynamic restores in progress.

DMSPOOLS

This file maintains the status of all tape volumes defined to various SAMS:Disk tape pools. Each volume is marked as being either an available scratch tape or an in-use tape.

DSNINDEX

This file contains an entry for each data set backed up or archived either EXPLICITLY or IMPLICITLY by the system. The records are keyed by data set name, and multiple versions of a given data set may be present.

MIGRECAT

An entry for each data set placed on the backup tape created during MIGRATION OF SEQUENTIAL data sets to tape is maintained in this file to facilitate recataloging all of the data sets should a primary tape be damaged or lost. SAMS:Disk does not provide any formal automated means for managing the entries in this subfile because there is no way of knowing how long to keep them. For this reason, this subfile will need to be purged occasionally to keep it from becoming excessively large.

RACFENCD

A record is placed in this file to cross-reference each RACF-protected data set and its associated SAMS:Disk-generated profile.

RETCMDS

A record is placed in this file for each RESTORE command to be processed on a deferred basis; this includes TSS and batch requests. The deferred command requests are eventually deleted subsequent to a batch DISPOSE job.

RETEXCLD

If sysparm RETEXCLD is specified with a value of Y, a record is placed in this file for each data set restored. An installation-defined grace period is assigned to each. An unexpired grace period prevents implicit archive from archiving the data set again.

Restrictions

A maximum of 12 subfiles is presently supported. However, you can limit this even further by specifying sysparm FILESCNT. For more details, please review the sysparm description for *FILESCNT* on page [150](#) in this manual.

Subfile Entry Definition

All subfiles to be used are defined to SAMS:Disk by entries in the FILEDEFN member of parmlib. During files initialization, the member entries are read and used to determine how the data set is to be formatted.

A sample set of subfile definition entries is distributed with the system as member FDSAMPLE in parmlib. To activate the definitions, you must create a member named FILEDEFN and copy in the sample values. An entry is required in the FILEDEFN member for each subfile of the files data set that will be used.

With the exception of the DSNINDEX subfile, the capacity values supplied in FDSAMPLE should be sufficient for a substantial period of time. All of the fields within FILEDEFN are described below:

Subfile Definition Parameters

The subfile definition parameters discussed here are found in dsn=dms.parm-lib(FILEDEFN). When you are defining subfiles, remember that the maximum number of subfiles that may be defined is 12.

Table 1-1. FILEDEFN Record Format

Positions	Field Name	Description
1-8	Subfile Name	The 8-character name of the subfile being defined
9-11	Logical Record Length	The 3-digit length of the records that will constitute the subfile.

Positions	Field Name	Description
12-14	Key Length	The 3-digit length of the record keys.
15		Character "C" to indicate capacity field follows.
16-23	Capacity (maximum record count) needed	The number of records the subfile must be able to index. This number is the theoretical maximum allowed; for example, if inserts were in the perfect sequence to fill every data block -- in other words, when totally "compressed", this number of records can be indexed. In the real world, inserts are random, resulting in "block splits" which cause embedded freespace (as much as 50 percent) in any given data block (or possibly all blocks). For this reason, it is recommended that you set the capacity value to roughly twice the number of records you estimate will be inserted.
24	Shared DASD	The letter Y in this column indicates the file can be accessed concurrently by multiple tasks.
25	No Duplicates Allowed	The letter "Y" in this column indicates that the file does not permit entries with duplicate keys.
26	Deferred Write	The letter "Y" in this column indicates that the nature of the processing of this file does not require that each record added be written back immediately. This flag should be specified as "Y" only for file name DASDSPCB. All others should be specified as "N".
27	File validate	The letter "Y" in this column indicates that information regarding the file structure is to be validated during file reload.

Files Data Set Management

Formatting the Files Data Set

After the FILEDEFN member has been created, the files data set can be created and formatted with the following FILEINIT. During file initialization, the FILEDEFN member entries are read and used to determine how the files data set is to be formatted. These FILEDEFN member entries are accessed only during file initialization, after which all information is carried within the data set itself. A Files Data Set Status report will be created upon completion of the FILEINIT JCL.

```
//FILEINIT EXEC PGM=ADSMI002,PARM=ADSDM100,REGION=512K
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//FILES DD DISP=(,CATLG,DELETE),DSN=SAMS.DISK.FILES,
// UNIT=SYSDA,SPACE=(CYL,10,,CONTIG),
// DCB=(DSORG=DA)
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

Figure 1-1. JCL to Create/Format the Files Data Set

The subfiles are formatted for optimum performance based upon the physical blocksize of the files data set. The physical blocksize of the files data set is the only value common to all of its subfiles. SAMS:Disk automatically assigns the optimum blocksize according to the device type being used. (You may override the default BLKSIZE by specifying a value in your JCL, but it is NOT recommended.) The major concern is to specify an appropriate value in the FILEDEFN entries.

Note: Please review the topic "*Files Data Set Capacity*" beginning on page [10](#) in this manual.

Files Data Set Status Report

The following JCL can be used to produce this report.

```
//STATUS EXEC PGM=ADSMI002,PARM=ADSDM107,REGION=512K
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

Figure 1-2. JCL for Files Data Set Status Report

This JCL produces a report that indicates the current usage for each subfile defined to the files data set. A critical field that should be inspected regularly is the OVER-

FLOW BLOCK COUNT. Whenever overflow blocks exist, the definition entry for each subfile with overflow blocks must have its record capacity increased. The entire files data set should then be reorganized immediately. Failure to do so will result in performance degradation and unpredictable results.

1994.029 JAN 29, 1994		FILES DATA SET STATUS								PAGE 1	
SATURDAY 11.38 PM										SAMS:Disk v.r.m	
DATA SET NAME WORKSHOP.FILES										TOTAL BLOCKS 300	
PHYSICAL BLOCK SIZE 27998										BLOCKS REMAINING 226	
BLOCKS IN USE 74										-----OPTION FLAGS-----	
SUB-FILE	RECORD	KEY	RECORD	MAXIMUM	BLOCK	REC PER	MAX	BLOCK	OVERFLOW	SHARE NODUP DEFER VALID GEN	
NAME	LENGTH	LENGTH	COUNT	RECORDS	LENGTH	BLOCK	BLOCKS	COUNT	BLOCKS		
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	
DSNINDEX	256	44	4180	254188	55808	218	1166	20	0	Y	Y
ARCHVOLS	110	6	7	710438	27940	254	2797	1	0	Y	Y
DMSPOLS	60	26	0	434312	27960	466	932	1	0	Y	Y
RETEXCLD	82	50	79	176638	27962	341	518	1	0	Y	
MIGRECAT	118	50	0	122766	27966	237	518	1	0	Y	
ARCHCMDS	164	45	0	96900	27880	170	570	1	0	Y	Y
RESTCMDS	164	44	0	96900	27880	170	570	1	0	Y	Y
DMSPARMS	100	8	0	650349	27900	279	2331	1	0	Y	
DASDSPCB	84	50	0	172494	27972	333	518	1	0		Y
RACFENCDC	90	49	0	163360	27900	310	527	1	0	Y	Y

Figure 1-3. Sample Files Data Set Status Report

Field Descriptions — Status Report

SAMS:Disk v.r.m:

The release number will appear where v.r.m is shown.

Physical block size:

The actual block size for the files data set. An optimum value for the device is automatically assigned by SAMS:Disk unless you specify a BLKSIZE value in the initialization JCL.

Total blocks:

The total number of physical blocks within the files data set. Some of these will contain data (blocks in use), and others will be "free blocks", available for record inserts (blocks remaining).

Blocks in use:

When initialized, the files data set is formatted from beginning to end with physical blocks of binary zeros (see *Total Blocks* field on page 8 in this manual). A minimum of one index block and one data block are then "allocated" to each subfile. All other blocks are "free" (see *Blocks Remaining* field on page 8 in this manual), and are subsequently allocated only when record inserts require more data blocks. The "blocks in use" number is the total number of physical blocks that have been allocated due to record inserts.

Blocks remaining:

The number of physical blocks initially formatted with binary zeros that have not yet had any records inserted. The number of unused, or free blocks in the data set.

Subfile name:

The name of either a SAMS:Disk or user-defined subfile.

Record length:

The length of the logical record within the named subfile.

Key length:

The length of the key portion within the logical record. For example, the key to the DSNINDEX record is the data set name, which is 44 bytes of the 256-byte record.

Record count:

The current number of records contained within the subfile.

Maximum records:

The maximum number of records that can be inserted and still indexed. This value is computed at initialization time from the capacity fields you specified in the FILEDEFN member of parmlib. When the actual record count exceeds half of this maximum count, you should consider specifying a higher capacity value in the FILEDEFN entry.

Block length:

Records for each subfile are grouped together to form a "logical" block that is also a "best fit" into a physical block (or multiples of physical blocks). Therefore the value displayed should be slightly less than a multiple of the physical block size.

Rec per block:

The number of records grouped together to form a "logical" block for the named subfile.

Max blocks:

The maximum number of logical data blocks that can be indexed for the subfile. Multiplying (max blocks)(rec per block) results in the maximum number of records that can be inserted and still indexed.

Block count:

The number of logical data blocks in use (indexed) for the subfile.

Overflow blocks:

When an additional logical data block is allocated to a subfile but there is no room left in the index to keep track of it, it is called an overflow block. Overflow blocks are chained from the relevant data blocks, and will cause considerable overhead or other unpredictable errors. The capacity for the subfile should be increased, and the files reorganized.

Option flags:

Share: Y indicates that integrity is maintained during concurrent access by multiple tasks.

Nodup: Y indicates that entries with duplicate keys are not allowed.

Defer: Y indicates that I/O can be buffered. This option requires special processing and is supported for only the DASDSPCB subfile.

Valid: Y activates internal validation routines when deemed appropriate.

Files Data Set Capacity

Estimating Capacity Requirements

There are two requirements for estimating files data set capacity. The first requirement is estimating the number of records the files data set should contain (FILEDEFN Specifications). The second requirement is understanding how much space (Physical space) should be allocated on the device based on the first requirement.

Estimating Subfile Capacity Requirements

Since the DSNINDEX subfile contains an entry for every data set backed up or archived, a beginning estimate is simply the number of data sets that are to be backed up with SAMS:Disk. If SAMS:Disk is to be used to back up all of your packs you need to answer three questions.

1. What is the total number of data sets on these packs?
2. How many versions of each data set do you want to keep?
 - a. Archives and/or Backups
3. How will my estimates comply to the Index Maintenance policies that manages what remains and/or is deleted from the files data set.

Estimating Physical Space Requirements

$$\text{Total Number of Tracks} = \frac{(\text{Subfile Capacity Requirements})}{(\text{Records per track})}$$

Performance Considerations

The files data set EXCP interface is designed to optimize performance at the logical block level. Normally, higher capacity requirements utilize larger block sizes yet the EXCP interface is designed to access these blocks in a single I/O, without performance penalty.

If your DSNINDEX subfile capacity requirement is substantially less than the maximum capacity requirement for the device type and you are experiencing a high number of block splits and/or performance degradation, use the maximum capacity requirement for the device.

Estimating F6425 Physical Space

If the files data set is to reside on a F6425, the following chart can be used as a guide in estimating the required physical space:

Table 1-2. Files Data Set Physical Space on a F6425

Subfile	Index Record Capacities	LRECL
DSNINDEX	4,470,240	256
ARCHVOLS	12,509,510	110
DMSPOOLS	7,646,005	60
RETEXCLD	3,109,563	82
MIGRECAT	2,159,962	118
ARCHCMDS	1,712,425	164
RETCMDS	1,712,425	164
DMSPARMS	11,740,767	100
DASDSPCB	3,035,681	84
RACFENCD	2,884,842	90

Minimum Allocation Requirements

Capacity Requested: 4,479,240 DSNINDEX subfile
 Number of recs/trk: 183
 Number of Tracks: 24,465*
 Number of Cylinders: 1,631*

F6425 Device Characteristics

Tracksize: 47,968
 Max BLKSIZE: 47,476
 Halftrack BLKSIZE: (23,476) X (2) / LRECL = Number of recs/trk

* Since logical data blocks are not assigned to a subfile until actually needed, similar to managing a PO data set, you don't need to allocate extra physical space until it is really needed. These numbers are based on each track or cylinder being full.

Table 1-3. F6425 Device Characteristics

MODEL	Cylinders	Trks/Cyl	Number of Trks
D	1,770	15	26,550
T	2,655	15	39,825

Note: Please review the topic "*Capacity Considerations*" beginning on page [16](#) in this manual.

Estimating F6427 Physical Space

If the files data set is to reside on a F6427, the following chart can be used as a guide in estimating the required physical space.

Table 1-4. Files Data Set Physical Space on a F6427

Subfile	Index Record Capacities	LRECL
DSNINDEX	6,374,376	256
ARCHVOLS	12,509,510	110
DMSPOOLS	7,646,005	60
RETEXCLD	3,109,563	82
MIGRECAT	2,159,962	118
ARCHCMDS	1,712,425	164
RESTCMDS	1,712,425	164
DMSPARMS	11,740,767	100
DASDSPCB	3,035,681	84
RACFENCD	2,884,842	90

Minimum Allocation Requirements

Capacity Requested: 6,374,376 DSNINDEX subfile
 Number of recs/trk: 218 records
 Number of Tracks: 29,240*
 Number of Cylinders: 1,949*

F6427 Device Characteristics

Tracksize: 58,786
 Max BLKSIZE: 56,664
 Halftrack BLKSIZE: (27,998) X (2) = Number of recs/trk

* Since logical data blocks are not assigned to a subfile until actually needed, similar to managing a PO data set, you don't need to allocate extra physical space until it is really needed. These numbers are based on each track or cylinder being full.

Table 1-5. F6427 Device Characteristics

MODEL	Cylinders	Trks/Cyl	Number of Trks
MOD 1	1,113	15	16,695
MOD 2	2,226	15	33,390
MOD 3	3,339	15	50,085

Note: Please review the topic "*Capacity Considerations*" beginning on page [16](#) in this manual.

Capacity Considerations

Block Splits

Since the logical block is kept in alphabetical order, but the inserts are in random order, an insert may require splitting a FULL logical block into two half-full blocks before the record can be inserted in its proper sequence.

Assume the worst case, that all blocks have been split and are only half full. If a subsequent series of updates requires inserts to the same block (names happen to be alphabetically adjacent), that block will need to be split also, but it cannot be. The insert will fail because all blocks are "in use", even though each is only half full. Notice that at this point, inserts are failing when only half of the theoretical capacity has been reached.

If the insert locations just happen to fall alphabetically into the correct blocks the theoretical capacity could be reached without block splits.

Overflow Blocks

If the logical indexing capacity is actually reached before it is noticed, processing will still continue as long as there is physical space left in the data set to allocate the needed data blocks. These non-indexed data blocks are called overflow blocks, which seriously degrade future processing and can cause unpredictable results. Messages are issued whenever this condition occurs. The Files Data Set Status report will indicate the overflow blocks. You should increase the capacity field for the subfile in the FILEDEFN member, and then use the REORG utility on the files data set as soon as possible.

Extending Allocation

When 100,000 bytes or fewer remain in the entire files data set, a message will be issued every time an entry is added. The message states that the files should be enlarged. This is done by unloading the current records (UNLOAD), and then re-loading (RELOAD) them into a data set with a larger space allocation. If these messages go unheeded, an appropriate message is issued and a user abend is issued when the physical space is exhausted.

Please review your policies concerning Index Maintenance. Mis-management of the files data set may impact capacity issues.

Files Data Set Utilities

Unload, Reload, And Reorg

Three utility functions are provided to backup and reorganize the files data set: UNLOAD, RELOAD, and REORG ALL.

- UNLOAD - unloads the files data set to a sequential data set to create a backup.
- RELOAD - reloads the files data set from the sequential data set.
- REORG - reorganizes the files data set by unloading the records to a sequential file, sorting them, and reloading them.

Selection of a subset of the files data set is permissible during both the unload and reload operations. Caution should be exercised, however, when specific subfiles are selected since the ENTIRE files data set is REFORMATTED EVERY time a RELOAD is done (that is, a RELOAD does not insert records at the logical subfile level leaving the other subfiles intact). All subfiles are reformatted and then the unloaded records are reloaded. Care must be taken to ensure that all subfiles with records to be saved are both unloaded and reloaded.

The unload/reload support will normally be used for one of two purposes. One role is to permit the allocation of a larger files data set to accommodate file growth. The new file capacities must be entered in the FILEDEFN member prior to executing the reload program. Since the reload program reformats the entire files data set, destroying the contents of all subfiles, every subfile must first be unloaded.

The other role is to re-order the data blocks and reduce block splits. The files data set itself may be large enough, but growth in one or more subfiles causes an increasing number of block splits which increases the number of I/Os performed.

Unload

JCL — UNLOAD Files

When you UNLOAD ALL subfiles and specify DAYS=(nn), indicating log data set retention, SAMS:Disk will dynamically allocate the sequential data set(s) into which the files will be unloaded. If it is present in the JCL, the SEQFILES DD is used. If you let SAMS:Disk dynamically allocate this file, your unload copy of the files data set will be called *"files.dsname.current-date.current-time"*. If the files data set name exceeds an allowable amount for the date appendage, truncation of the files data set name will take place at position 27 of the files data set name. Note that the enqueue on the files data set is held for the entire UNLOAD process. When the UNLOAD is complete, "B" log files (review the topic *"Naming the Log Data Sets"* on page 22 for details) older than the retention period specified by parameter

DAYS= will be deleted. When the sequential data set is dynamically allocated, the expiration date used for the data set is taken from sysparm DYNEXPDT.

"I" log files, those created by PFD (review the topic *"Naming the Log Data Sets"* on page 22 for details) are not selected for deletion at this time, as they may represent recovery data for more than one files data set. To delete these log data sets, select them using the DSCL CREDIT criteria, and then issue the archive or delete action command.

The SIMULATE parameter is provided with the UNLOAD function of SAMS:Disk. When specified, all reports and messages will be generated, but the actual UNLOAD will not take place until the SIMULATE parameter is removed from the command.

The unloading of the contents of the files data set from disk to tape (a sequential data set on disk works equally well) is accomplished with the following JCL. A DSN=any.gdg(+1) is suggested.

```
//UNLOAD EXEC PGM=ADSMI000,PARM=ADSDM177
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SEQFILES DD DISP=(NEW,CATLG,DELETE),
//          DSN=SAMS.DISK.SEQFILES.UNLOAD,
//          DCB=(LRECL=268,BLKSIZE=5900,DSORG=PS,RECFM=VB),
//          SPACE=(6144,(360,360),RLSE),UNIT=SYSDA
//SYSIN DD *
UNLOAD ALL,FILES=,SIM,DAYS=,UNIT=,PRI=,SEC=
```

Figure 1-4. Unload Files JCL

UNLOAD Command and Parameters

```
UNLOAD ALL,FILES=,SIM,DAYS=,UNIT=,PRI=,SEC=
```

ALL

The presence of the parameter causes ALL subfiles to be unloaded.

FILES=

A list of one or more subfile names may be specified to confine the output to a subset of the files data set. Omission of this OPTIONAL parameter results in all files being unloaded.

SIM

This parameter causes the output reports to be produced as if processing had been performed; however no data sets are unloaded or deleted.

DAYS=

This parameter specifies the retention period for SAMS:Disk-generated recovery log data sets. Log data sets older than the specified number of days will be deleted following completion of the UNLOAD. SAMS:Disk will not delete batch log data sets that are less than 1 day old. If DAYS=0 is entered, SAMS:Disk processes the log data sets using DAYS=1. This prevents SAMS:Disk from deleting log data sets that are currently being built. No processing against archived log data sets is done.

UNIT=

This is a list of names for the units allocated for the UNLOAD data set when the unload is considered a backup (ALL is specified and the DAYS= parameter is used.) If the unit is a disk device, the esoteric name should be used. If the unit is a tape device, TAPE, F6470, DYN1, DYN2 or DYN3, should be entered and the DYNxUNIT sysparm values are used.

PRI=

This is the primary space allocation amount in cylinders for the UNLOAD data set. The default amount is 1 cylinder.

SEC=

This is the secondary space allocation amount in cylinders for the UNLOAD data set. The default amount is 1 cylinder.

RELOAD**JCL — RELOAD Files**

The RELOADING of an unloaded set of files to the files data set is accomplished with the following JCL. Care should be taken to ensure that the sequential input file contains all of the needed files, since the reload reformats the entire files data set prior to doing the reload.

```
//RELOAD   EXEC PGM=ADSMI002,PARM=ADSDM192
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//FILES    DD DISP=SHR,DSN=SAMS.DISK.FILES
//MSGPRINT DD SYSOUT=A
//PARMLIB  DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SEQFILES DD DISP=OLD,DSN=SAMS.DISK.SEQFILES.UNLOAD
//SYSIN    DD *
RELOAD ALL,FORMAT,MODULES,FILES=,FREESP=
```

Figure 1-5. JCL to RELOAD the Files Data Set

RELOAD Command and Parameters

RELOAD ALL,FORMAT,MODULES=,FILES=

ALL

This parameter causes all of the unloaded records to be reloaded.

FILES=

A list of one or more subfile names may be specified to confine the load to a subset of the records on the sequential storage medium. Omission of this parameter results in all records from the sequential data set being loaded, regardless of the subfile from which they came.

FORMAT

This parameter causes the files data set to be entirely reformatted prior to loading the records from the specified subfiles.

MODULES=

A list of one or more names of modules to be given control in order to inspect or alter each record reloaded.

FREESP=

By default, RELOAD processing reloads subfile records into each logical data block, until the maximum record limit is reached. After which a BLOCK SPLIT is required to insert other entries into the logical block. Specifying this parameter with a percentage value helps alleviate block splits. This parameter is limited to only the DSNINDEX subfile.

For example, a value of 20 allows 20 percent of free space for each logical data block.

REORG Subfiles

If any of the subfiles of the files data set go into overflow blocks and the capacity increase and reorganization are not done (see *"Files Data Set Status Report"* on page 7 in this manual), the possibility exists that new inserts or deletes may cause the proper key sequence for some records to be DESTROYED. No records are lost, but the improper key sequencing can cause unpredictable results. If this condition occurs, the reorganization utility may be used to reorder all subfile records into their correct sequence.

```
//REORG EXEC REORG
//SYSIN DD *
        RELOAD ALL,FORMAT
```

The RELOAD command as described in "Reload Command for Files Data Set" on the previous page may be used to restrict the reload to a subset of the files.

A message is printed during the REORG to indicate when it starts reloading the files. If the system abends after this message, a RESTART procedure has been provided to complete the reload. The RELOAD command is used during the RESTART PROCESSING.

```
//RESTART EXEC RESTART
//SYSIN DD *
        RELOAD ALL,FORMAT
```

Files Data Set Logging

Activation

To activate the files data set logging feature, you must specify `sysparm FILOGNAM` with the high-level node of the data set name that will be used for the logs. The default value for `FILOGNAM` is blanks, which disables the logging feature. `FILOGNAM` must be placed in member `SYSPARMS` (described on page 562 in this manual) and must not be overridden in individual job steps. This ensures that all SAMS:Disk processing affecting each files data set will have a log associated with it.

For more information, please review the sysparm description for *FILOGNAM* on page 150 in this manual.

To direct the size and destination of the files data set logs, see the sysparm descriptions of *FILESPEC* and *FILEUNIT* beginning on page 150 in this manual.

Naming the Log Data Sets

SAMS:Disk generates data set names for the log data sets as they are created. The high-level node of the data set name is the value you specify for sysparm FILOGNAM. To this node SAMS:Disk will append a second-level node of I or B, indicating how the log was generated:

- I — indicates that the log was generated from an PFD interactive request, or from a restore request (i.e., explicit batch restore, or a DMSAR auto-restore)
- B — indicates that the log was generated from a batch job, with the exception of the explicit restore

This is followed by a date/time stamp generated during dynamic allocation of the data set. One batch log data set will be created for each SAMS:Disk job run, but only one PFD log data set will be created daily.

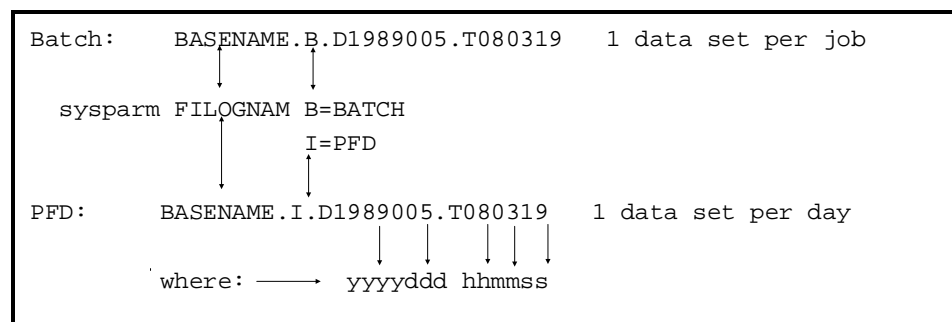


Figure 1-6. Files Data Set Logging DSNNAME

Format of the Log Data Sets

In the batch log data set, the first record contains the name of the Files Data Set. In the PFD log data set, each log record is preceded by the files data set identifier record. This enables SAMS:Disk to support logging and recovery to multiple files data sets.

Managing the Logs

With the logging function activated, SAMS:Disk will create and accumulate a large quantity of log data sets over time. The UNLOAD utility can be run with a DAYS= parameter to delete old files, and retain only those log data sets created more recently than the number of days specified on the parameter. This feature only works with the ALL parameter specified. Running UNLOAD with a list of files specified will cause the processing of the log data sets to be bypassed.

Recovery

Because the files data set is a critical resource, SAMS:Disk also provides a means to recover the files data set should it be lost or corrupted. To perform the recovery, SAMS:Disk optionally creates a log recording all activity, including batch jobs and PFD activity.

SAMS:Disk provides the ability to recover the files data set should it become lost or damaged. To use the files data set recovery utility, you must first activate the files data set logging feature. The recovery utility uses records from the log data sets as input, and updates the appropriate backup copy of the files data set to achieve the forward recovery.

The FRECOVER command of the files data set recovery utility initiates files data set recovery. The input to FRECOVER is the audit log data sets created during files data set logging. Review the topic "*Activation*" on page [22](#) for more information.

In the FRECOVER command, you specify the files data set to be recovered and the audit log data sets to be selected. Audit log data sets are selected based on: time and date parameters, and the high level qualifier name of the audit log data sets in the sysparm FILOGNAM. The audit log records from the audit log data sets selected are used to cause forward recovery of the files data set.

Running the Utility

Before you run this utility in live mode, run it in simulate mode to verify that the parameters are correct and that the correct logs are being selected.

To recover a lost files data set, run the FRECOVER utility with the following:

- The most recent backup version of the files data set, from the last UNLOAD job run used to backup the files data set.

- The audit log data sets from the Files Data Set Logging starting from the date and time after the last UNLOAD, and ending at the current time and date.

To recover a damaged files data set, run the FRECOVER utility with the following:

- The most current UNLOAD of the files data set prior to the damage.
- The audit log data sets which contain the subfile records needed to update the files data set, excluding the log where the damage occurred.

Internal Processing

The files data set recovery utility first selects the audit log data sets within the time and date parameters specified in the FRECOVER command. The time and date parameters should not proceed the date and time of the last files data set backup. Next, it looks inside these data sets and selects the records that were created for the files data set to recover, and that are within the time and date parameters. If any of the audit log records in a batch audit log data set do not fall within the specified time and date, the data set is bypassed.

The selected audit log records are sorted by the files data set subfile name and the audit log record time and date. The subfile name, subfile function and subfile record (which are the original parameters used to update the files data set) are extracted from the audit log record. These subfile parameters are then used to update the files data set.

Security Considerations

The utility checks the user access to the subfile you want to recover before it updates the subfile, if you have subfile security set up at your installation. However, the user access to any of the data within the subfile is not checked, such as the data set name within the DSNINDEX record. We recommend that you protect the FRECOVER command using the SECURCMD sysparm and the SECURDSN member of parmlib to prevent unauthorized use of this utility.

JCL — Files Data Set Recovery Utility

The following JCL is used for the forward recovery of the files data set.

```
//FRECOVER EXEC FRECOVER
//SYSIN DD *
FRECOVER FILES=filesdsn,BTIME=hhmmss,BDATE=date,
ETIME=hhmmss,EDATE=date,SIMULATE
```

Figure 1-7. Files Data Set Recovery Utility JCL

FRECOVER Command and Parameters

`FRECOVER SIMULATE=, FILES=, BTIME=, BDATE=, ETIME=, EDATE=`

SIMULATE

Specify this parameter to select the audit log data sets and audit log records which would be used to update the files data set if running in live mode. A report will be printed.

FILES

This parameter is required to specify the files data set to be recovered.

BTIME

This parameter is required to specify the begin time of the audit log data sets and audit log records to be selected for files data set update. It must be in hhmmss format, hh=hour, mm=minutes, and ss=seconds.

BDATE

This parameter is required to specify the begin date of the audit log data sets and audit log records to be selected for files data set update. This may be in any accepted SAMS:Disk format.

ETIME

This parameter is required to specify the end time of the audit log data sets and audit log records to be selected for files data set update. It must be in hhmmss format, hh=hour, mm=minutes, and ss=seconds.

EDATE

This parameter is required to specify the end date of the audit log data sets and audit log records to be selected for files data set update. This may be in any accepted SAMS:Disk format.

Sample - Files Data Set Recovery Report

1989.192	JUL 11, 1989	F I L E S D A T A S E T R E C O V E R Y				PAGE	1
TUESDAY	10.10 AM					SAMS:Disk	8.2
AUDIT LOG DATA SET SELECTION REPORT							
FILES DATA SET = LABS.KSV.FILES							
DATA SET NAME							

LABS.LOG.KSV.B.D1989174.T125102							
LABS.LOG.KSV.B.D1989174.T125642							
LABS.LOG.KSV.B.D1989174.T131735							
LABS.LOG.KSV.B.D1989181.T083227							
LABS.LOG.KSV.B.D1989181.T084519							
LABS.LOG.KSV.B.D1989181.T090218							
LABS.LOG.KSV.B.D1989186.T144516							
LABS.LOG.KSV.B.D1989186.T152416							
LABS.LOG.KSV.I.D1989192.T094730							
1989.192	JUL 11, 1989	F I L E S D A T A S E T R E C O V E R Y				PAGE	2
TUESDAY	10.10 AM					SAMS:Disk	8.2
AUDIT LOG RECORD UPDATE REPORT							
FILES DATA SET = LABS.KSV.FILES							
SUBFILE	FILES	DMS	SUBFILE	NEXT 50 BYTES OF SUBFILE			
NAME	FUNC	FUNCTION	KEY	STARTING AFTER SUBFILE KEY			

ARCHVOLS	SR	ARCHIVE	@99952	WRK003 &	SBKSV.DISKPRIM.DMSC.D19891		
ARCHVOLS	SR	ARCHIVE	@99952	WRK003 &	SBKSV.DISKPRIM.DMSC.D19891		
ARCHVOLS	SR	ARCHIVE	@99950	WRK003 &	SBKSV.DISKPRIM.DMSC.D19891		
ARCHVOLS	SR	ARCHIVE	@99950	WRK003 &	SBKSV.DISKPRIM.DMSC.D19891		
ARCHVOLS	SI	DMS	@99997	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SR	DMS	@99997	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SI	DMS	@99996	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SR	DMS	@99996	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SI	DMS	@99995	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SR	DMS	@99995	WRK003 &	PROD.DMS.ARCHPRIM.DMSC.D19		
ARCHVOLS	SR	IXUPDATE	@99950	WRK003 &	SBKSV.DISKPRIM.DMSC.D19891		
ARCHVOLS	SR	IXUPDATE	@99949	WRK001 &	SBKSV.DISKCOPY.DMSC.D19891		
DSNINDEX	SF	DMS	LABS.KSV.MULTIV5	CAT002	-	@99997	
DSNINDEX	SF	DMS	LABS.KSV.MULTIV5	CAT002	-	@99996	
DSNINDEX	SF	DMS	LABS.KSV.MULTIV5	CAT002	-	@99995	
DSNINDEX	SF	DMS	LABS.KSV.MULTIV5	CAT002	-	@99999	
DSNINDEX	SF	DMS	LABS.KSV.MULTIV5	CAT002	-	@99998	
RESTCMDS	SD	DERASE	LABS.KSV.LOG2	SBKSVRES	; ;	LABS.KSV.LOG2	

Figure 1-8. Sample Files Data Set Recovery Report

Subfile Dump Utility

JCL — Subfile Dump Utility

A batch hex dump of any record from any subfile within the files data set can be achieved by running the following JCL.

```
//DUMP      EXEC PGM=ADSMI002, PARM=ADSDM613, REGION=1024K
//STEPLIB   DD DISP=SHR, DSN=SAMS.DISK.LOAD
//ABNLDUMP  DD DUMMY
//CMDPRINT  DD SYSOUT=*
//FILES     DD DISP=SHR, DSN=SAMS.DISK.FILES
//MSGPRINT  DD SYSOUT=*
//PARMLIB   DD DISP=SHR, DSN=SAMS.DISK.PARMLIB
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSIN     DD *
FILEDUMP FILE=subfilename,KEY=keyofrecord,VERSION=versionnumber
```

Figure 1-9. Subfile Dump Utility JCL

Subfile Dump Command for the Files Data Set

The command to dump one or more of the files is as follows.

FILEDUMP FILE= KEY= VERSION=

FILE=

This parameter is required and must specify any one of the SAMS:Disk subfiles.

KEY=

This required parameter is used to specify the key of the record desired. Turn to page [31](#) for information on identifying the keys of the individual subfiles. See the *DSNINDEX* Record Format on page [529](#) to find out what the keys consist of.

VERSION=

This is an optional parameter, used to specify up to the 99th version of a data set in the *DSNINDEX* subfile. This parameter is acceptable only for the *DSNINDEX* subfile. If no version parameter is specified, a default of 0 is used. The version parameter is expressed as a positive number from 0 to 99, with 0 indicating the most current version of the data set.

Sample Subfile Dump Utility

A sample of the command and the output produced by the command can be seen below.

FILEDUMP FILE=DSNINDEX,KEY=LABS.HANEY.CNTL,VERSION=1

```
94.007 JAN 07, 1994          SUBFILE DUMP UTILITY
FRIDAY 3.52 PM

-----
DSNINDEX RECORD DUMP
DEC  HEX      HEXADECIMAL DATA      CHARACTER DATA
0000  0000 D3C1C2E2 4BC8C1D5 C5E84BC3 D5E3D340 *LABS.HANEY.CNTL*
0016  0010 40404040 40404040 40404040 40404040 *          *
0032  0020 40404040 40404040 40404040 D3C1C2E2 *          LABS*
0048  0030 F8F10729 00005401 55540164 02000E80 *81..... .*
0064  0040 000F0015 00000033 00000256 0003A358 *..... *
0080  0050 0008E2C2 F2F4F0F3 0C00          *..SB2403.  . *
```

Figure 1-10. Sample Subfile Dump Utility

Merging Files Data Sets

The previously described RELOAD ALL / REORG procedure is designed to process a single files data set. On some occasions it may be desirable to merge the contents of two or more files data sets, or to perhaps resequence a single files data set from its unloaded backup copy. The following describes the requirements of such an operation.

1. Ensure that all of the ARCHVOLS records being merged have unique keys; that is, the same key does not appear in more than one files data set. Having the same tape in more than one files data set should be exceedingly rare, but if archiving to disk has been done using more than just one of the files being merged, duplicate keys are probably present. In any case, you must identify all tape and disk keys that appear in more than one of the files.

You may either purge (with SCRATCH=NO) these duplicates from each files data set, or keep one of them and purge the others. In either case, the objective is to arrive at a status where all keys are unique, before proceeding. When the following steps are completed, use the rebuild function to add them back into the new files data set. The duplicate keys for the disk archives are eliminated because the rebuild function will generate new and unique keys.

2. The DSNINDEX subfile requires different sorting fields from all other subfiles. Therefore, unload the DSNINDEX records from each files data set to a sequential data set of its own. The command for the files unload utility would be:

```
UNLOAD FILES=(DSNINDEX)
```

3. Sort these sequential data sets into a single output data set using the following sort control fields.

```
SORT FIELDS=(5,8,CH,A,13,44,CH,A,67,3,CH,D,63,4,CH,D)
```

4. Unload the other subfiles from each files data set to a new sequential data set.

```
UNLOAD FILES=(a list of all files except DSNINDEX)
```

5. Sort the non-DSNINDEX records into a single output data set using the following sort control fields.

```
SORT FIELDS=(5,8,CH,A,13,50,CH,A)
```

6. Concatenate the sorted output from steps 3 and 5 and use as input to the RELOAD utility program, using the command:

```
RELOAD ALL,FORMAT
```

7. Execute the LISTD command and visually verify that the new files data set is in correct sequence.

8. As mentioned in item 1, use the rebuild function to reinsert records for any duplicate ARCHVOLS keys that were purged.

Common File Handler Interface

Subfile records in the files data set may be accessed by user- written modules to either read or write records in the same manner as SAMS:Disk modules that interface with the file handler.

Records in the files data set may be referenced both by user exit modules invoked by SAMS:Disk and/or user programs that are run independently of SAMS:Disk. It is only necessary that the file handler be called via the MI interface module that is shipped with the system. For best results, all user programs and modules should be compiled using the High Level Assembler. This release of SAMS:Disk was compiled using version 2.0 of the High Level Assembler, and using Assembler H, especially without slacmods, may cause unpredictable results.

It should be noted, however, that most SAMS:Disk modules are neither reentrant nor reusable. When programming for interactive applications, more detailed knowledge of program internals will be required to ensure proper execution.

A unique record, identified by supplying the key to the record, may be retrieved from a subfile within the SAMS:Disk files data set by executing a call in the following manner. If the subfile permits duplicate keys, a request to retrieve a unique record will return the first (most recently added) record for the supplied key.

```

MVC DSNKEY,=CL144'PAY.MASTER'
* Retrieve the data set index record
CALL MI, (ADSDM095,RU,FILENAME,DSNREC,RESULT),VL
CLC RESULT,=H'1'
BH NOTFOUND

*
ADSDM095 DC CL8'ADSDM095' file handler control module
RU       DC CL2'RU'      service type = retrieve unique
FILENAME DC CL8'DSNINDEX' subfile name to access
DSNREC   DS 0CL256
DSNKEY   DC CL44' '      key of record to retrieve
          DC CL212' '    data from retrieved index rec
RESULT   DC H           result

```

Figure 1-11. Call to Retrieve Data Set Index Record

File Handler Interface

```
CALL MI, (ADSDM095, FUNCTION, FILENAME, KEY&RECAREA, RESULT), VL
```

Where:

ADSDM095

Is always the 8-character name of the File Interface Control module. ADSDM095 DC CL8'ADSDM095'

FUNCTION

A 2-character mnemonic specifying the type of service to be performed.

FILENAME

The 8-character name of the logical subfile to be accessed. It must be in the file definition entries.

KEY& RECAREA

A work area for record placement, the length of which is specified in the corresponding file definition entry. For unique retrieval of a record, the key of the desired record must be placed in the high-order portion.

RESULT

The halfword return code set by the file handler to reflect the conclusion of the requested service.

Function service types

RU - RETRIEVE UNIQUE

This service will return the first occurrence of one or more records having the key that is left-justified in the work area. The length of the key is indicated in the file definition entry.

1. requested record returned
2. key not found

The logical position within the file for the purpose of sequential retrieval when a return code of 2 is passed is the point at which the requested record would have been.

Other than the first occurrence of a given key may be uniquely retrieved by appending to the key in the record area a suffix that consists of a number in parentheses. The number specifies the desired relative occurrence of the designated key where the most recent version is 0.

RN - RETRIEVE NEXT

The next sequential record will be returned by use of this service. It is required that the key for the previous service be present in the record area for maintaining logical position.

To retrieve the first record in a file, it is necessary that the key portion of the record area be initialized with spaces prior to this type of call.

1. next record returned
2. end of file

SI - STORE INSERT

A record is added to the file with the use of this service. Duplicate keys are permitted based on the file definition, with the most recently inserted record being presented first on sequential retrieval.

1. record inserted
2. insert of duplicate keyed record not allowed

SR - STORE REPLACE

A record is updated by use of this service after it has been retrieved. However, a record may be replaced without a preceding retrieval. If you are accessing a subfile that allows duplicate keys, you should position to the correct record before doing the SR function. You can position to a particular record by using the RU function with a relative record number in parentheses or by using the RU function to position to the first record and using the RN function interactively until positioned correctly.

1. record replaced
2. record not found

SD - STORE DELETE

This service is provided for elimination of records from a file.

1. record deleted
2. record not found

There is no requirement for a record to be retrieved prior to a delete call.

CL - CLOSE THE FILES

This service closes the files data set and frees all associated buffers. There are no other parameters for this option. (The other service requests will automatically (re)open the files if needed.)

DW - DEFERRED WRITE BUFFER PURGE

This service applies only to the DASD billing subfile DASDSPCB. Service requests to modify records in this subfile are kept in memory until a different data block is to be updated. Therefore update programs must do a final call using this option to write out the last updated buffer. The record area and result parameters are ignored, if specified, and may be omitted for this service.

For example:

```
CALL MI, (ADSDM095,DW,DASDSPCB),VL
```

Interface for Sequential Reads

SAMS:Disk also provides an alternative to a series of RN calls to the common file handler interface, which provides much **higher performance** for reading a large number of records sequentially. This high-performance file handler interface keeps each block of records in a buffer until all records have been returned. Once the block has been read into the buffer, this interface will not see any updates done to the block on disk by the ADSDM095 interface.

The intent of this faster interface is for programs that list or scan the files data set, without concern for intermixed updates. The structure of this interface's parameter list is identical to that just presented. However, the interface module is ADSDM463 instead of ADSDM095, and only two function types are valid: RN and RP.

```
CALL MI, (ADSDM463,FUNCTION,FILENAME,KEY&RECAREA,RESULT),VL
```

Where:

ADSDM463

Is always the 8-character name of the File Interface Control module. ADSDM463 DC CL8'ADSDM463'

FUNCTION

A 2-character mnemonic specifying the type of service to be performed, either RP or RN.

RP — REPOSITION

Use this service to start a series of RN calls. It resets internal pointers so that you may provide a different key (that is, a new starting location) in subsequent series of read next (RN) calls. Any other parameters specified on an RP call are ignored.

RN — RETRIEVE NEXT

This service will return the next sequential record. To start a series of RN calls, you must provide the key of the starting location in the KEY&RECAREA parameter of the first call. Any key value, including SAMS:Disk pattern-matching characters, may be used to establish a starting location. A key of spaces identifies the starting position as the very

first record in the file. This initial call will return the first record in the KEY&RECAREA parameter.

If you do not alter the key at this point, subsequent RN calls will read the following records in ascending key sequence. If you wish to start reading another series of records, issue a reposition (RP) call, then provide the key for the new starting position in the KEY&RECAREA parameter in the first RN call in the series. Continue the series with the remaining RN calls.

Return Code	Meaning
1	next record returned
2	end of file

Processing Support

In the event that an invalid service or file name is passed to the file handler, it will abend the task with a user code of 0016 after printing an appropriate message.

To facilitate the reuse of DASD space within the files data set, a free queue of data blocks is maintained for each subfile. Blocks are assigned to this queue when they contain no more records. Conversely, when a block split occurs during record insertion, the free queue will be inspected before additional space is acquired from the common free area.

When an insert attempt that requires a new data block is made, but both the free queue chain for the subfile and the common area free queue are exhausted, the task will be abended with an appropriate message. Prior to such termination, warning messages will be produced each time an allocation to a subfile occurs when the residual free space falls below the 100,000- byte threshold.

A warning message will appear each time a block split occurs for a file and no capacity remains in the index block. This condition does not render the respective file inoperative, but will result in additional overhead when referencing blocks without index entries. The condition is remedied by reorganizing the file after updating the maximum record count field in the file definition entry for the appropriate subfile.

Support Modules and Functions

ADSDM095 — COMMON FILE HANDLER INTERFACE

This module accomplishes the following:

- Reads in FCB (FCB entries for all subfiles).
- Locates proper FCB entry for requested subfile.
- Identifies requested service type.
- ENQ/RESERVE issued for all non-deferred store/update requests.
- Calls subcontrol modules to perform requested service.
- DEQ/RELEASE issued for all non-deferred store/update requests.

ADSDM463 — INTERFACE FOR SEQUENTIAL READS

This *High Performance* module accomplishes the following:

- ADSDM106 — SUBCONTROL MODULE
 - Reads in index block(s) for requested subfile unless
 1. Subfile is using deferred writes (DW)
 - or
 2. Sysparm INDXREADN is specified
 - Calls appropriate service level module.
- ADSDM087 — SI, STORE INSERT

Note: This is the only service type that can cause block splits.

ADSDM088 ... SR, STORE REPLACE
 ADSDM089 ... SD, STORE DELETE
 ADSDM096 ... RN, RETRIEVE NEXT
 ADSDM097 ... RU, RETRIEVE UNIQUE

Service Level Supporting Modules

ADSDM083 — Physical DASD block processor

OPTIONS = open; close; read; write; blksize; dsname

ADSDM081 — Read logical (subfile) block

ADSDM082 — Write logical (subfile) block

ADSDM580 — Locate record by key

Note: This module checks for a deferred write (DW) subfile being processed. If the input key maps to a data block not currently in the buffer and if the current buffer has been updated (write pending flag is on), it purges/writes out the updated block and then reads in the new block needed for the specified key. Since an updated block is held in memory for perhaps several updates, protection against loss of data due to a concurrently running task cannot be provided and the ENQ/RESERVE are not done.

ADSDM086 — Block split handler

Note: A block split is caused by a record insert into a full block. A free block is obtained from the free chain off the subfile's index block or, if none exists there, then from the free area controlled by the allocation control record. The "top half" (high key portion) of the existing block is moved to the "free" block. The record insert is then made to the appropriate upper or lower half-block.

ADSDM084 — Construct keyed file index block

ADSDM079 — Allocate free space control module

ADSDM093 — Validate the file control blocks

This module follows all data block chains and:

- Verifies block count
- Checks for loops in the block pointers
- Checks for logical EOFs
- Checks for ascending keys
- Checks that index key equals block key
- Verifies overflow block chains

Chapter 2. Tape Management

This section describes the following topics regarding the management of archive/backup tapes:

- Allocation Considerations (page 38 in this manual)
- DDNAMES Used (page 38 in this manual)
- Dynamic Allocation Versus Supplied JCL (page 38 in this manual)
- Producing Duplicate Archive Tapes Concurrently (page 39 in this manual)
- Specifying Blocksizes for Archive/Backup Tapes (page 39 in this manual)
- Creating Multivolume Tape Data Sets (page 40 in this manual)
- Running Archive/Backup Functions in Simulate Mode (page 41 in this manual)
- Running Multiple Archive/Backup Jobs Concurrently (page 41 in this manual)
- Cataloging the Tape Archives (page 41 in this manual)
- Tape GDG Support (page 42 in this manual)
- Assigning Tape Expiration Dates (page 42 in this manual)
- Tapepool Considerations (page 43 in this manual)
- Tape and Cartridge Switching — Length Considerations (page 47 in this manual)
- Tape Format (page 49 in this manual)

Additional information related to managing tapes can be found in the following:

User's Guide

The following sections of the *User's Guide* should be reviewed:

- Files Data Set IXMAINT section (begins on page 291 in the *User's Guide*)

- Archive Index Rebuild Utility (begins on page 317 in the *User's Guide*)
- MERGE section (begins on page 325 in the *User's Guide*)

Installation Guide

The following section in the *Installation Guide* should be reviewed:

- Customizing the SAMS:Disk Tape Management Support (page 62 in the *Installation Guide*)

Systems Guide

The following section in the *Systems Guide* should be reviewed:

- Archive/Backup Considerations (page 57 in this manual)

Allocation Considerations

DDNAMES Used

SAMS:Disk uses the following DDNAMES for tape and cartridge units:

- //ARCHIVE0 — to create the primary archive tape/cartridge
- //ARCHIVEC — to create the duplicate copy tape/cartridge
- //ARCHIVEN — to create the primary merge output tape/cartridge
- //ARCHIVCn — to create the duplicate copy merge tape/cartridge
- //ARCHIVES — to read an archive F619 tape (i.e., restore or merge)
- //ARCHIVER — to read an archive cartridge (i.e., restore or merge)

Dynamic Allocation Versus Supplied JCL

In the MSP environment, SAMS:Disk will dynamically allocate tape units to the above DDNAMES whenever they are needed, whether for output tapes being created or input tapes to be read. The types of units dynamically allocated are controlled by system parameters ARC0TYPE, MERPnTYP, ARCCTYPE and MERCnTYP. For information on these sysparms, turn to the topic "*Specifying the Archive Medium*" on page 58 in this manual.

Supplying the tape dd statements in the JCL will override dynamic allocation. However, dynamic allocation does provide the following advantage. If the job will not actually use the tape units (for example, a simulate run, no data sets are selected to

be archived, or the job is limited to reporting functions), the tape units will not be allocated, and unneeded waits are avoided. It also keeps the units free for use by other tasks.

Since dynamic allocation often calls for a different tape unit address for each allocation, you should also review the use of sysparm DYNUKEEP with a value of Y. This causes a tape unit to be allocated when needed, but then kept and reused throughout the remainder of the job step, rather than the system allocating and freeing a tape unit for each tape that is needed. Reusing the same tape unit may benefit the tape operators, but it also increases run time. Before the next tape can be mounted, SAMS:Disk is forced to wait for the prior tape to be rewound. Dynamic allocation of a different unit allows some of the processing to be overlapped.

Alternatively, you may also supply the tape dd statements in the JCL (they exist as comment statements in the distributed JCL procedures). This also ensures that the needed tape units are available and allocated before the job starts. If the units are not available when a dynamic allocation request is made, the operator will be prompted for the necessary units.

Note: When supplying the tape dd statements in JCL or sysparms ARCONAME, ARCCNAME, MERPnNAM, or MERCnNAM, make sure that the value you specify for the data set name does not exceed 22-characters. This will allow SAMS:Disk to append the required date/time at the end of the ARCHVOL data set. For related information, review the *Figure* on page 61 and the sysparm description for *ARCTNAME* on page 128 in this manual.

Producing Duplicate Archive Tapes Concurrently

When you run archive or backup jobs, two copies of archive volumes are produced concurrently unless you nullify the allocation of the //ARCHIVEC dd statement. To suppress the duplicate copy, specify sysparm ARCCTYPE with a value of NULL, or provide the dd statement in your JCL as //ARCHIVEC DD DUMMY.

The volume serial number of a duplicate copy tape is carried in the archive volume record for the primary tape. The duplicate copy tape also has an archive volume record built for it. If another copy is made (the triplicate copy), the volume record for the duplicate will point to the triplicate. In each case, the volume record points to the copy record, which points to the next copy, and so on until the end of the chain is reached, indicated by blanks in the volume serial field for the next copy tape.

In the event that a primary archive tape is damaged or otherwise becomes unavailable, the RESET command can be used to flag it as disabled, and cause subsequent RESTORE or MERGE requests to automatically call for the duplicate copy backup tape.

If you produce an archive volume without a duplicate copy, it is possible to create a copy after the fact, just as you can produce the triplicate from the duplicate. See *"Making Additional Copies of Archive Tapes"* described on page 322 in the *User's Guide*.

Specifying Blocksizes for Archive/Backup Tapes

The following rules govern the blocksize value used for the archive data sets, whether they are on tape or disk.

1. If archive tapes are being dynamically allocated, a blocksize of 32,760 is used. To use a smaller blocksize, supply the dd statements in your JCL with the blocksize as desired.
2. When archiving to disk, the most efficient blocksize is assigned by SAMS:Disk based upon the device type being used. You may override this value by specifying sysparm ARCDKBZ.
3. If either the primary copy or the duplicate copy is on disk, the blocksize for the copy on disk dictates the blocksize used for the tape.
4. The blocksize for the primary copy dictates that of the duplicate copy, if both are on tape.
5. When copying user data sets from disk to the archive medium, SAMS:Disk will split user data blocks across archive data blocks (similar to the concept of "spanned records"). This means that the blocksize of the archive medium may indeed be less than the blocksize of a data set being archived.

Creating Multivolume Tape Data Sets

If an extremely large data set on disk is to be archived (or backed up), or if very short tapes are mounted, several tapes may be required to hold the data set. SAMS:Disk permits a data set to span as many as 35 volumes, and automatically allows for this when dynamically allocating the tapes. But if you provide the tape dd statements in your JCL, you must also allow for up to 35 volumes. This is done by adding the volume count parameter, VOL=(,,35), to your JCL.

EDRC Support in F6470 Cartridge Devices

EDRC is a feature of the F6470 cartridge subsystem. It provides data compression at the cartridge subsystem level. This feature can be controlled when creating output tapes with SAMS:Disk. Refer to the sysparm description for *IDRCUSED* on page 152 for more information. EDRC is activated during OPEN processing and can be used with either JCL or dynamic allocation processing.

When the EDRC feature is not available, two bits in the JFCTRCH field of the JFCB are defined as zeros. When the EDRC feature is available, the two bits indicate that compaction is on (x'08') or off (x'04'). Just prior to open the bits are checked. If both bits are zero (EDRC not available), the IDRCUSED sysparm is ignored and nothing is done. If the bits are x'08' or x'04' (EDRC available), the IDRCUSED sysparm is checked and if it is specified Y or N the bits are altered to reflect the setting of the sysparm.

When an Archive tape is created with the EDRC feature active, the ARCHVOL record for this tape volume is flagged to indicate so. At restore time, no user action is required because compaction is automatically detected.

Considerations for the use of the EDRC Feature

- Do not use the EDRC Feature for any data targeted for a disaster recovery site unless the recovery site also has EDRC installed. If the recovery site does not have EDRC installed, the data is not recoverable.
- Examine the DCDATACP sysparm and run tests turning SAMS:Disk compression on and off, both with and without EDRC compaction. You might want to compare both the amount of compaction and the run time difference to determine the optimum settings for these sysparms.
- During duplex processing, if one copy is to a device without the EDRC feature you may not get an efficient use of your cartridge, since primary and copy cartridges must switch simultaneously.

Running Archive/Backup in Simulate Mode

When you run archive or backup functions in the simulate mode, the tape drives do not need to be allocated. If you are using dynamic allocation options, the tapes will not be allocated. If you are supplying the archive dd statements in your JCL, you may dummy them out for simulate mode.

Running Multiple Archive/Backup Jobs Concurrently

It may be desirable to execute multiple archive or backup jobs concurrently on either the same or different CPUs in order to reduce the elapsed time needed for the process. If these jobs are to be run on the same CPU, or if you have a cross-system enqueue facility in place, the DSNNAME parameter on the //ARCHIVEC and //ARCHIVE0 dd statements must be different for each concurrently executing job. This prevents the standard DSNNAME enqueue lockout that will occur if the same name is used with DISP=NEW in more than one job.

By default, SAMS:Disk generates unique names (see *"Naming the Archives"* on page 61 in this manual), and dynamically allocates them to these two ddnames, avoiding the enqueue lockout. If you select the option of using non-unique names during backup or archive processing, you must manually supply different DSNAMES for each job by either putting them in the JCL or by overriding sysparms ARCONAME and ARCCNAME.

Each of your concurrent backup jobs should also process a list of volumes different from the other jobs. It is also advantageous if these volume lists achieve as much channel separation as possible in order to reduce channel contention among the jobs.

Cataloging the Tape Archives

As previously described, if you want to catalog your archive tapes, specify a value of C for sysparm ARCTNAME. If you also use a security system, please heed the authorization requirement documented in the sysparm description for *ARCTNAME* on page 128 in this manual.

To uncatalog tapes that are expired by MERGE or IXMAINT, specify sysparm UNCATARC with a value of Y.

Tape GDG Support (No Longer Recommended)

Prior to Release 7.2, users desiring to catalog archive tapes had to use GDG support. Although the GDG support is no longer needed, it may still be used by specifying sysparm ARCTNAME with a value of G. This option is not recommended; it is maintained strictly to provide compatibility with prior releases.

The following paragraphs document how this option works. This explanation is provided strictly for existing users who do not choose to change to the enhanced (default) method of generating tape archive data set names, and as such is not applicable to new users.

GDG support may be invoked by SAMS:Disk to generate unique tape archive data set names. This is done by appending the GDG name to the base name specified in sysparms ARCONAME (for the primary copy) and ARCCNAME (for the duplicate copy). This generation name is also kept synchronized with the archive volume index. Two special sysparms are provided for this support:

1. ARCTNAMEG must be specified to synchronize the creation of index entries and catalog entries.
2. UNCATARCY is used to cause a tape whose index entry has been deleted in MERGE or IXMAINT to be uncataloged.

If ARCTNAMEG is specified, a generation index for the appropriate number of entries must first be built into the supporting catalog. Since SAMS:Disk creates a generation number for each tape created, the number of entries is equal to the maximum number of archive tapes at any given time. Once this generation index level is established, it must be specified as the data set name on all of the dd statements without an appended level number or absolute generation number. SAMS:Disk will build the correct level number and catalog the tape. The JCL disposition parameter for the archive tape(s) should always be coded as DISP=(NEW,KEEP,KEEP).

If a duplicate copy tape is being produced, a generation index must also be built for it. The default index is DMS.ARCHCOPY, but you may use any name you desire in sysparm ARCCNAME for dynamic allocation or supply it in the JCL.

Assigning Tape Expiration Dates

Expiration dates may be assigned to archive tapes either explicitly or implicitly. An explicit expiration date is created by hard-coding its value in sysparm DYNEXPDT for dynamic allocation or by specifying the EXPDT= or RETPD= parameter in the JCL for the //ARCHIVE0 dd statement.

The implicit assignment of an expiration date to the archive tape volume is accomplished by finding and using the highest expiration date that is assigned to any individual data set being archived. SAMS:Disk does this by determining what data sets are to be processed, assigning each its archive retention period, and then internally queuing the archive request. The value of the retention period assigned is supplied by sysparm ARCRETPD for explicit archives, sysparm RETRETPD for backup/archive, or parameters RETPD= and EXPDT=, which may be supplied on the ARCHIVE or RETAIN command. When all data sets to be processed have been scanned, the highest retention date is known. This date is then used as the expiration date for the archive tape, and is recorded in the archive volume index. The internally queued requests are then processed.

SAMS:Disk will honor any non-zero expiration date and place it in both the tape label and in the archive volume index record. Hard-coding an expiration date, however, does create the possibility of expiring archive tape volumes prior to the expiration of all archived data sets on the archive volume.

We recommend using an explicit expiration date of "99365" on your archive tapes. This is considered a "never scratch" date by SAMS:Disk and protects against the previously mentioned case of a tape being written over before all of its data sets have expired. This is the technique called for by the distributed JCL procedures and system parameter default.

If you have implemented tape management support, it is recommended that all archive tape dd statements use the default EXPDT=99365. SAMS:Disk will notify the tape management interface as soon as any given tape can be returned to scratch status. See the *"Customizing the SAMS:Disk Tape Management Support"* on page 62 of the *Installation Guide* for more information.

Tapepool Considerations

The SAMS:Disk tapepool support provides you with the ability to define up to 30 different pools of tapes. (You may define 30 pools in parmlib member POOLDEFS. However, this member is also used to define disk archive pools. The sum of tape and disk pool definitions cannot exceed 30.) Each pool name is associated with (mapped to) a data set name on an archive dd statement. Thus, if all archive tapes have the same name or prefix, one pool is sufficient. However, if the duplicate copy tapes have different names than the primary tapes (as in the default), it is possible to define a separate pool of tapes for each. Similarly, backup tapes can be separated from archive tapes by giving them different names and associated pools.

The association (or mapping) of pool names to archive data set names is made by entries in member POOLDEFS in the parmlib data set. Each entry consists of an eight-character pool name (of your choice) and the tape data set name. (Remember, the data set names are generated from sysparms ARCONAME and ARCCNAME or are provided in the JCL.) Each entry must be enclosed in quotes, with one or more blanks between the pool name and data set name. For example:

```
'ARCHPRIM DMS.ARCHPRIM/'           pool for primary archives
'ARCHCOPY DMS.ARCHCOPY/'           pool for duplicate archives
                                     or perhaps
'ARCHPOOL DMS.ARCH/'               pool for all archive tapes
'BKUPPOOL DMS.BACK/'               pool for all backup tapes
```

The first step in establishing tapepool support is to decide how many different data set names (categories of tapes) you have a real need for. One suggestion is to separate your archive from your backup tapes, and if you create duplicate copy tapes, to separate them from the primary tapes as well. Once decided, make sure that the names you have chosen are provided to SAMS:Disk via overriding sysparms for each job, or in the JCL for each job. Then create member POOLDEFS in your parmlib data set with the appropriate entries. You may copy in member SAMPDEFS as a starting point.

The next step is to place scratch tapes into the pools in preparation for use. This can be done most easily from the PFD Online Tape pool Management facility. If you do not have this support, you must use the *Tapepool Update Utility* documented on page 46 in this manual. The PFD facility and the *ADD Command* (described on page 46 in this manual) also provide the means to do any maintenance functions that may be required, such as listing, adding or deleting entries.

At this point you are ready to activate the support. This is done by adding sysparm TAPEPOOLy to member SYSPARMS in the parmlib data set. Once this is done, new scratch tapes to be used for archive, backup or merge functions will be located in the appropriate pool, and their status updated to show that they are in-use. When subsequent merge or index maintenance functions determine that no unexpired data sets remain on a tape volume, its status is updated to indicate that it is available again for use as a scratch tape.

When a group of tapes are entered into a tapepool, they are entered at the bottom of the current list of scratch tapes for the named pool. Tapes that are scratched by MERGE or IXMAINT processing are also put at the bottom of the tapepool scratch list, in the order in which they are scratched. Scratch tapes needed by ARCHIVE, RETAIN, or MERGE are taken from the top of the scratch list. This ensures that the tapes are cycled, much like tape drives are cycled. Due to the complex logic that MERGE and IXMAINT use to scratch SAMS:Disk archive tapes, the tapepool scratch list quickly loses its initial VOLSER order. This causes no problems for SAMS:Disk, however. The tapepool LIST command described below can be used to inspect the status of the tapepools.

Tapepool support is also designed to maintain the integrity of the pools in the following manner:

1. The same volume cannot be placed into more than one pool.
2. Only the PFD or batch utility program will add a volume into a pool; that is, scratch tapes mounted by operators or expired tapes being "returned to scratch status" by index maintenance are never added to a pool. The volume must already exist in the pool such that SAMS:Disk merely updates its status.
3. When creating new output tapes with the pool support active, a valid pool name to use must be found or processing terminates.

Tape volumes may be supplied by any combination of the following three techniques. It should be noted that these methods are progressive, in that processing begins with the first that is applicable and continues until either no more tapes are needed or the operator is unable to supply an acceptable volume serial.

From JCL:

If tapepool support is being used, from 1 to 5 volumes may be listed in the VOL=SER= parameter of the //ARCHIVE0 and //ARCHIVEC dd statements. These volumes are used before accessing the defined tapepool.

For non-tapepool support, up to five volumes may be supplied on the VOL=SER= parameter of the //ARCHIVE0 and //ARCHIVEC dd statements. If more than five volumes are required, the operator will be requested to mount scratch tapes.

From a tapepool:

If the tapepool support is being used, volume serial entries are taken from the pool when the VOL=SER= parameter is not specified in the JCL, or when the JCL list of volume serials has been exhausted.

From the operators:

If the tapepool support is being used but the pool is out of scratch tapes, a WTOR message is issued to allow the console operator to enter the volume serial of the next scratch tape to be used.

It is permissible to CANCEL an archive run when a volume serial cannot be supplied in response to the WTOR message, since the processing for any volume written up to that point is complete.

Tapepool Updates from PFD

The Online Tapepool Management facility supplied with the SAMS:Disk PFD support provides the easiest means of listing and maintaining your tapepools. In addition to being able to add or delete entries, sysparm TPOOLCHG is available to

allow an "I" to be placed to the left of an entry to change it from "scratch" to "in-use", or an "A" to change an "in-use" status to "available" (scratch).

These options should be used with extreme caution, however, since the tapepool must also be kept synchronized with the current ARCHVOLS listing of "in-use" tapes. For related information, please review the *REPARCH Utility* beginning on page 342 in this manual.

Tapepool Update Utility

If the Online Tapepool Management facility supplied with the SAMS:Disk PFD support is not available, the following batch utility provides an easy means of listing and maintaining your tapepools. The available commands and there functions are:

- LIST - generate a volume status list for a requested pool.
- ADD - add one or more volumes to a pool.
- DELETE - delete a specific volume from a pool
- ERASE - remove all volumes from a specified pool.

```
//TAPEPOOL EXEC DMSPOOL
//SYSIN      DD  *
LIST  POOLNAME=,VOLUME=,STATUS=
ADD   POOLNAME=,VOLUME=,NUMBER=,STATUS=
DELETE VOLUME=
ERASE  POOLNAME=
```

Figure 2-1. Sample JCL to Execute Tapepool Utility

LIST Command

LIST POOLNAME=,VOLUME=,STATUS=

POOLNAME=

Specify one or more eight-character pool names. (A maximum of 25 entries supported.)

VOLUME=

This optional parameter specifies one or more volume serials in the tape pool. The status report is restricted to just the named volumes. (A maximum of 25 entries supported.)

STATUS=

Specify "S" if you wish to list only scratch volumes, or "U" for in-use volumes. The default is to list both scratch and in-use volumes.

ADD Command

ADD POOLNAME= , VOLUME= , NUMBER= , STATUS=

POOLNAME=

Specify the eight-character name of the tape pool into which volumes are to be added.

VOLUME=

Specify the volume serial to be added to the named pool.

NUMBER=

Specify a value of from 1 to 32767 for this optional parameter to indicate the number of volumes to be added, beginning with the volume specified. The numeric portion of the volume serial is incremented by one, the new volume serial is added to the pool, and the process repeated until the requested number of tapes has been inserted. For example, to add volume serials DMS001 through DMS025, specify:

```
ADD POOLNAME=TAPEPOOL , VOLUME=DMS001 , NUMBER=25 ,  
STATUS=S
```

STATUS=

This optional parameter specifies the status of the volume(s) to be added. Use "S" for scratch volumes and "U" for volumes in-use by SAMS:Disk. It defaults to S, and normally you should never have a need to insert in-use tapes.

DELETE Command

DELETE VOLUME=

VOLUME=

The specified volume will be deleted regardless of the pool it is in.

ERASE Command

ERASE POOLNAME=

POOLNAME=

The eight-character pool name specified will have all of its volumes deleted. The pool ceases to exist.

Tape and Cartridge Switching — Length Considerations

SAMS:Disk was designed to create single volume ARCHVOLS whenever possible. Before a data set is written to an ARCHVOL, SAMS:Disk attempts to predict whether or not the data set will fit on the current ARCHVOL. If the prediction indicates that the data set won't fit, the ARCHVOL is closed and a new one is opened. This design minimizes the need to mount more than one tape during subsequent RESTORE operations.

Note: The only exception is when the initial data set on an ARCHVOL exceeds the capacity of the ARCHVOL. Only then will SAMS:Disk create a multi-volume ARCHVOL, spanning up to 35 volumes if necessary.

The method in which SAMS:Disk calculates its predictions is determined by the output media used, and the values associated with four (4) system parameters. Also factored into the calculation is whether or not software and/or hardware compression is being utilized.

The following figure summarizes the media options available (with the exception of DISK), and which system parameters are in effect:

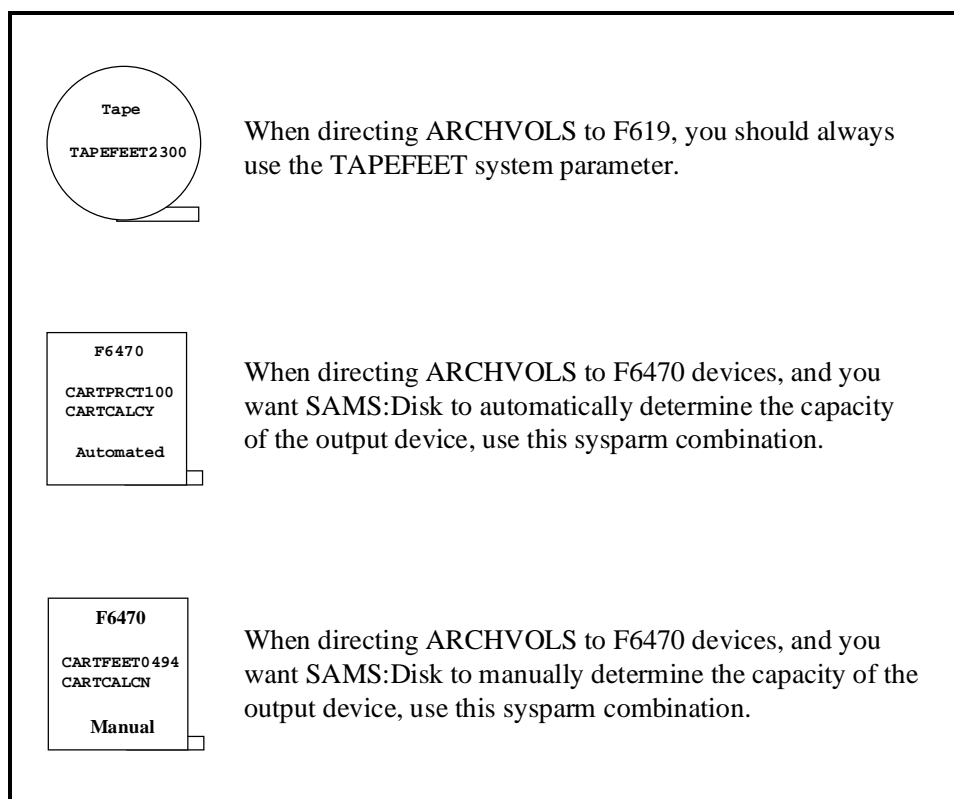


Figure 2-2. Predicting Output Device Capacities

For TAPE (F619) Devices

When determining if the *next* data set will fit onto the current ARCHVOL, SAMS:Disk factors in the value specified for TAPEFEET, along with a number of other variables such as:

- Bytes
- Blocks
- Block Size
- Recording Density
- Size of Inter-Record Gaps

The result is then added to the number of bytes and blocks previously written to the ARCHVOL. If the data set will fit, SAMS:Disk begins transferring the data. If the data set will not fit, SAMS:Disk closes the ARCHVOL and begins transferring the data onto a new ARCHVOL.

For related information and a detail description of the TAPEFEET system parameter, please turn to page [190](#) of this manual.

For Cartridge (F6470) Devices

When determining if the *next* data set will fit onto the current ARCHVOL, SAMS:Disk factors in the value specified for CARTFEET, along with a number of other variables such as:

- Bytes
- Blocks
- Block Size
- Recording Density
- Size of Inter-Record Gaps

The result is then added to the number of bytes and blocks previously written to the ARCHVOL. If the data set will fit, SAMS:Disk begins transferring the data. If the data set will not fit, SAMS:Disk closes the ARCHVOL and begins transferring the data onto a new ARCHVOL.

For related information and a detail description of the CARTFEET system parameter, please turn to page [133](#) of this manual.

Tape Format

The data archived is stored on standard labeled tapes as a sequential data set in undefined record format. The specific SAMS:Disk record types are presented in the topic "*Archive Tape Format*" on page [535](#) in this manual.

Chapter 3. Tailoring Options

General Information

SAMS:Disk provides two primary means of tailoring the system to more specific needs. You should become familiar with the capabilities each provides so that you may find relatively simple solutions to conditions that are unique to your installation.

1. SAMS:Disk provides for system parameters (sysparms) to be specified in member=SYSPARMS of the parmlib data set. The contents are retrieved at execution time and processing paths are varied accordingly. Default values for all sysparms are generated internally. You need to create member=SYSPARMS and specify a sysparm value ONLY IF YOU WANT TO CHANGE A DEFINED DEFAULT. The *Sysparms* section, beginning on page 103, provides a complete list of the sysparms and their defaults. There is no need to read and review every one of them until an actual need is made evident from your use of the various functions. Some of the more commonly used ones, or ones that are more important to consider in advance, are presented here to acquaint you with their use.
2. User exits are also supported. A complete list of these and their defined capabilities may be found in the *User Exits* section, beginning on page 213 in this manual.

In addition to sysparms and user exits, the following topics are highlighted in this section:

- Performance Tips (page 55 in this manual)
- Archive/Backup Considerations (page 57 in this manual)
 - Activating Data Compression (page 58 in this manual)
 - Specifying the Archive Medium (page 58 in this manual)
 - Using Multiple Types of Media (page 59 in this manual)
 - Naming the Archives (page 61 in this manual)
 - JCL Overrides and Data Set Name Usage (page 62 in this manual)
 - Assigning an Expiration Date to the Archive Data Set (page 62 in this manual)

- Archiving to Disk: Requirements and Recommendations (page [62](#) in this manual)
- Using Archive Functions to Manage Active Data (page [67](#) in this manual)
- Archiving the Archives is Not Allowed (page [68](#) in this manual)
- Archive Integrity and Security (page [69](#) in this manual)
 - Archive Tape Protection (page [69](#) in this manual)
 - Checks Against Overwriting Valid Tapes (page [69](#) in this manual)
 - Tape Status After Abends (page [70](#) in this manual)
 - Archive Disk Protection (page [71](#) in this manual)
- Processing PDSs that Contain Anomalies (page [71](#) in this manual)
- PFD Custom Reports (page [82](#) in this manual)
- User-Specified Condition Codes (page [87](#) in this manual)
- DASD Threshold Manager (page [90](#) in this manual)

Suggested System Parameters

System Parameters for Reports:

The Data Set Utilization report can be produced in three different sequences. By default it is produced in sequence by date-of-last-use. The following options are available:

Table 3-2. DSUTIL Report Sorting Options

Sysparm	Description
DSUTILSQ	In data set name sequence
DSUTILCF	By date-of-last-use within index

Both the explicit and the implicit archiving functions can produce reports in one to three sequences simultaneously. By default they are produced in both data set name and data set name within volume sequences. See sysparm *ARCHSORT*, described on page 127, to select a different option.

The sequential migration to tape function has this same ability. See sysparm *MIGRSORT*, described on page 159, to adjust its defaults.

The SMF report requires its input from a sequential data set. If your SMF records reside in a VSAM cluster, add a Fujitsu KQCAMS utility step to REPRO the records into a sequential data set to be passed to SAMS:Disk.

System Parameters for Messages

The following sysparms affect the number of informational messages that are printed for their respective functions, and should be kept in mind when initially testing and setting up SAMS:Disk runs.

Table 3-1. Message Suppression Sysparms

DSCLMSGGS	DSCL Data Set Selection Messages
RLSEDIAG	Idle Space Release messages
MIGBYPAS	Sequential Migration to Tape messages
VCBYPASS	Move/Copy messages

System Parameter for Lines Per Page

The number of lines per page on SAMS:Disk—generated output can easily be changed from its default of 58. For details, turn to the sysparm description for *RPTLINES*, located on page 180 in this manual.

System Parameter for Separator Pages

The number of separator pages can be changed or suppressed via sysparm *FLYCOUNT*, described on page 150 in this manual.

User Exit for All Sysout Print Lines

For more extensive monitoring or modifications of printed output, such as special routing for certain messages, consult the description of user exit *SYSOUTEX*, located on page 286 in this manual.

System Parameter for Files Data Set Logging

Three sysparms provide support for the files data set logging capability.

Table 3-3. Files Data Set Logging Activation Sysparms

Sysparm	Description
FILOGNAM	for specifying the name of the log data sets and to activate the feature.
FILEUNIT	for specifying the unit type allocated for the log data sets.
FILESPEC	for specifying the number of blocks allocated to the log data sets.

System Parameters for Archive/Backup

The following sysparms are commonly specified by many users and should be reviewed for applicability in your installation:

Table 3-4. Common Archive/Backup Sysparms

Sysparm	Description
ARCBLKSI	for processing data sets that have BLKSIZE=0
ARCDSORG	for processing data sets with unknown DSORGs
ARCEMPTY	for processing empty data sets
ARCMODEL	for processing model DSCBs
ARCRETPD	change retention period for explicit archives
RETRETPD	change retention period for implicit archives

System Parameters for Restore/Recover

By default, SAMS:Disk will issue a catalog locate for each data set being explicitly restored, unless a specific target volume is given. For volume recovery functions, these catalog locates are bypassed. Another locate is done in preparing the restore report. Both restore and recovery tasks will normally skip any preallocated version of a data set. Sysparms to review in order to change these defaults are:

Table 3-5. Common Restore/Recover Sysparms

Sysparm	Description
RESCHCAT	for volume selection for non-VSAM during Restore
RECCHCAT	for volume selection for non-VSAM during Recover
PREALLOC	controls whether or not to overwrite preallocated data sets

Performance Tips

Report Processing

Review the sysparms for the reporting function listed earlier that control the use of the LOCATE macro. This macro causes a high level of system overhead. Elapsed time values will be reduced significantly if the locate function is suppressed for runs that process a high number of data sets.

PDS Compress

Sysparm IOTRACKS also affects PDS directory processing. Tests have also shown that for directories in the range of one track or less (which is 46 directory blocks or less on a F6425), setting IOTRACKS to 1 can improve performance. If tests verify this in your environment, you should supply this value as a permanent sysparm override to the PDS Compress function.

Previous recommendations to exclude SMP data sets still being kept as PDSs (rather than VSAM data sets) are no longer applicable. The revisions allow all partitioned data sets to be processed equally well.

Files Data Set

Since the files data set contains the archive indexes, it is an active and critical data set when doing implicit archiving, backup, or any index maintenance function. ENQ and RESERVE macros are also issued against the data set and its containing volume. Improper selection of the containing volume may result in performance degradation of SAMS:Disk or other jobs trying to access the volume.

For important information, please review the sysparm description for *RSUPPRES* on page [181](#) in this manual.

I/O Buffering and Memory Requirements

A sysparm that can have a dramatic effect on SAMS:Disk performance and the amount of memory that will be required is IOTRACKS. IOTRACKS controls the maximum number of tracks read or written at one time by the SAMS:Disk EXCP access method, which is used for all but exception case handling of non-VSAM data sets. Review the sysparm description for *IOTRACKS* on page [154](#) in this manual.

Ideally, optimum performance is achieved when IOTRACKS allows a cylinder to be processed by a single I/O request; for example, 15 tracks for F6425 devices. The default value for IOTRACKS accomplishes this. You should test different values of this sysparm to find the best to suit your environment. As with most performance issues, there may be several factors to consider, such as the memory constraints, I/O contention, and page and swap activity.

First consider the size of these buffers themselves. A 15-track buffer on a F6425 is $(15) \times (47,476)$ or 712,140 bytes. SAMS:Disk double buffers, which means 1,424,280 bytes (or roughly 1.4 megabytes). For the Move/Copy function, which is reading and writing to disk at the same time, this means 2.8 megabytes merely for basic I/O data buffers. Memory is required for many other items as well, such as space for the loaded programs themselves and buffers for reading VTOCs, PDS directories, and control information from the SAMS:Disk parameter library. Many of these memory tables reside above the 16 MB line. A five- to six-megabyte region will usually be sufficient for all SAMS:Disk jobs, including Move/Copy. If real memory is very limited, however, this region requirement may be too high, and must be lowered by reducing the value of sysparm IOTRACKS.

Buffers are acquired based upon the values specified, but a single I/O request never exceeds a cylinder in size or crosses a cylinder boundary. If the number of tracks in a data set (or technically the number of tracks in each extent) is less than a cylinder in size, part of the buffer will always remain unused (wasted). It follows that if the average data set size in tracks is less than the value specified for the appropriate sysparm, you may set the sysparm to a lower value to reduce memory constraints without noticeably increasing the number of I/Os (which would degrade performance).

If the amount of available real memory is limited, having this sysparm set to a high value can cause jobs to be swapped frequently and/or paging rates to be high, which will degrade performance instead of improve it.

If the "to" and "from" volumes are on the same channel when using the Move/Copy function to relocate data sets, the job may perform better by setting IOTRACKS to a much smaller value, perhaps even 1. This is due to the inherent channel contention in this particular case, combined with the paging and swapping problem described above. Again, testing within your environment will demonstrate whether this case warrants special treatment.

Another sysparm to review when considering memory requirements is *IOMAXREC*, described on page 153 in this manual. This sysparm sets the maximum allowable record size for data sets to be processed by SAMS:Disk. It is used to acquire buffer space to contain the records during backup and data compression/decompression. The higher this value is set, the greater the memory requirements. It is recommended that this sysparm be set to the lowest value possible that will still allow SAMS:Disk to process. The default value is 65,000.

Archive/Backup Considerations

SAMS:Disk permits numerous tailoring options for archiving and backup functions. Some of the more important options to consider are discussed in this section.

Activating Data Compression

As an optional feature, SAMS:Disk can compress data as it is being written to the archive or backup data sets, whether they are on tape or disk.

If you activate data compression, the default technique is technique number 0. SAMS:Disk-supplied technique number 0 will typically reduce the space used by 40 to 50 percent, which will yield the same reduction in the amount of tape used. However, if you archive to disk instead of tape, the disk savings achieved may be closer to three or four to one, or even much higher. This is due to the optimized storage techniques in the archives, as well as wasted space in the original data sets.

As a direct result of compression, the number of I/Os to the archive media is also reduced by the same percentage, which may provide a significant improvement if I/O contention is a problem in your installation.

To achieve these benefits, however, you should also expect the CPU time to increase. For more information, turn to the topic *"Data Compression/Decompression"* beginning on page 448 in this manual.

The following sysparms are used by data compression. Review them for applicability in your installation:

Table 3-6. Data Compression Sysparms

Sysparm	Description
DCDATAACP	turns data compression off or on
DCCOMPTC	indicates the compression technique number
DCCMPEXn	exit for compression technique n
DCDCPEXn	exit for decompression technique n
DCDSNDEX	user exit for deciding if data compression is to be done for each data set, and what technique is to be used.
DCEXCTBL	exclusion table name
DCINCTBL	inclusion table name (overrides DCDATAACP"N")
DCRTSTAT	turns compression statistics off, on, or on with details for each data set compressed.

Specifying the Archive Medium

By default, SAMS:Disk dynamically allocates both the primary and duplicate copy to tape. The DD statement used for the primary is //ARCHIVE0 and for the duplicate copy is //ARCHIVEC. Sysparms ARC0TYPE and ARCCTYPE allow you to change the defaults from tape to either disk or F6470 cartridge units. Specify these sysparms with a value of DISK, TAPE or F6470, and place them in the SYSPARMS member of the parmlib data set. There are 2 ways to indicate that no duplicate copy is to be made:

1. Specify "NULL" to sysparm ARCCTYPE as "NULL"
2. In JCL, specify "DUMMY" to the //ARCHIVEC DD statement

For example, to direct the primary copy to disk and the duplicate copy to a F6470 cartridge unit, specify ARC0TYPEDISK and ARCCTYPEF6470.

You may also specify the archive medium through JCL. If tape or F6470 devices are specified via JCL, the devices allocated at job initiation will be used throughout the run; that is, the device will not be deallocated at the end of each tape, but only at the end of the job. (The dynamic allocation option can be made to behave in much the same way by specifying sysparm DYNUKEEP.) The only exception is when more than five volumes are needed to contain a single archive data set, in which case dynamic allocation is used.

You may also specify archival to disk in JCL by allocating any disk device to the //ARCHIVE0 DD statement. For example:

```
//ARCHIVE0 DD UNIT=SYSDA,VOL=SER=anyvol,DISP=SHR
```

Notice that this only indicates to SAMS:Disk to use disk archive. The volume (or data set) named on the DD statement has no meaning to SAMS:Disk, and dynamic allocation of the ARCHVOL will still be done.

Using Multiple Types of Media

This section describes how SAMS:Disk can utilize multiple types of media (i.e., F6470s, 3490Es, SILOs, etc.), sysparms to use, EDRC considerations, data compression, and MERGE processing.

Selecting the Output Media

The user selects the media SAMS:Disk will write to by specifying a value to the following sysparms, which are often entered as overrides in SAMS:Disk jobs, or kept in the SYSPARMS member in PARMLIB.

Table 3-7. Output Media Sysparms

Function	Primary SYSPARM	Copy SYSPARM	Notes
ARCHIVE	ARC0TYPE	ARCCTYPE	
BACKUP	ARC0TYPE	ARCCTYPE	
MERGE	MERPnTYP	MERCnTYP	Refer to the <i>LIMITS</i> parameter on page 336 of the <i>User's Guide</i> for detail information.

The following table shows the media options that can be specified with the above sysparms. The primary and its copy can be directed to different media. These two output media types are synchronized so that if one is shorter than the other and fills up, both are closed and new tapes are mounted and/or new disk archive data sets are allocated. Other sysparms that are in effect when the respective media option is used are shown below:

Table 3-8. Available Values for Media Option Sysparms

Media Option	Type of Device	Unit for Writing or Reading	Unit for Auto-Restore	Tape Length
TAPE	"Round" tape (default)	DYNTUNITtape	ARESUNITtape	TAPEFEET2300
F6470	F6470 cartridge	DYNCUNITF6470	ARESUNICF6470	CARTFEET0494
DYN1	Special device type 1	DYN1UNIT	ARESUNI1	TAPEFEET2300
DYN2	Special device type 2	DYN2UNIT	ARESUNI2	TAPEFEET2300
DYN3	Special device type 3	DYN3UNIT	ARESUNI3	TAPEFEET2300
DISK	DASD volumes	n/a	n/a	n/a
NULL	Disables Copy sysparms	n/a	n/a	n/a

Specifying the ESOTERIC Unit Name

The media options shown above relate to sysparms that provide the esoteric unit name SAMS:Disk uses to allocate the proper device for writing, reading and auto restoring. These sysparms are placed in the SYSPARMS member in PARMLIB and should not be changed unless the esoteric name changes, or a media type has been totally removed from SAMS:Disk control. The DYNn sysparm provides the flexibility of defining additional device types to SAMS:Disk that may be installed at various sites.

When a tape is created, SAMS:Disk stores the device type in the ARCHVOLS record along with a flag if the DYN1, DYN2, or DYN3 option was used. With this information, SAMS:Disk retrieves the esoteric unit name from the DYN_UNIT or ARESUNI_ sysparm so the proper device can be allocated.

Naming the Archives

In theory, the names you assign to archive data sets are of no consequence to SAMS:Disk. SAMS:Disk will keep track of the archive data sets regardless of their names. However, other issues dictate that naming conventions be observed, including the Fujitsu operating system and tape management concerns. SAMS:Disk generates unique archive data set names, providing flexibility in managing the archives and allowing them to be cataloged. Unique data set names also eliminates contention during dynamic allocation that would occur if non-unique data sets were used.

SAMS:Disk sysparm defaults generate unique archive data set names. The base names are taken from either sysparm ARCONAME or ARCHIVE0 DD statement (for the primary copy) and sysparm ARCCNAME or ARCHIVEC DD statement (for the duplicate copy). If both copies are being produced, the base names should be different — such as DMS.ARCHPRIM and DMS.ARCHCOPY.

Specify up to 22 characters for the base name and SAMS:Disk will generate a 22-character appendage as follows:

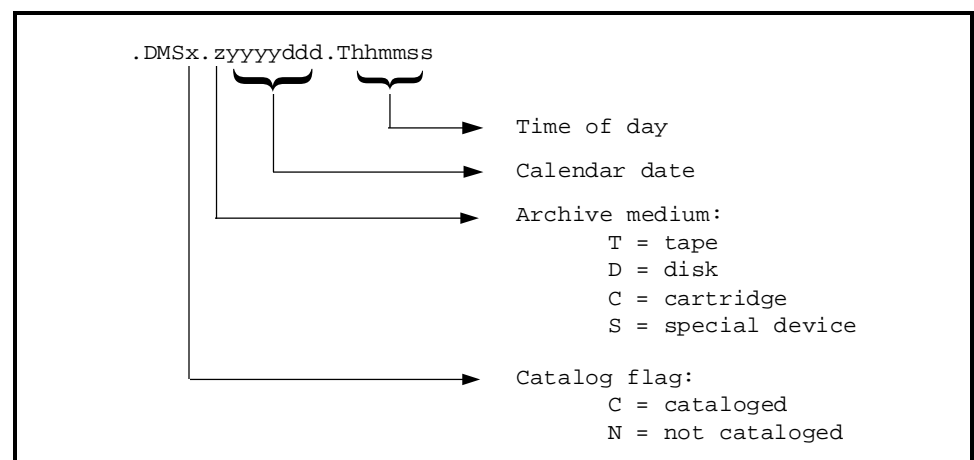


Figure 3-1. Generated 22 Character DSNAME

The calendar date above is the current date in Julian format, and the time of day is in hours/minutes/seconds. Hence the archive data set name provides the date and time it was created, and will always be unique.

For disk archive data sets, this generated name is always used and is always cataloged. For tape and cartridge archive data sets, the default is to use the generated name but not to catalog it. You may tell SAMS:Disk to catalog the tape or cartridge archive data set name by specifying a value of C for sysparm ARCTNAME.

JCL Overrides and DSNNAME Usage

As just explained, SAMS:Disk defaults to using dynamic allocation for the archive data sets. For tape and F6470 devices, however, you may override dynamic allocation by supplying the needed information on the appropriate DD statements in the JCL. SAMS:Disk will create the data set name from the DSNNAME you provide in the JCL unless you omitted it completely or specified a temporary name (DSN=&&anyname). In both of these cases, SAMS:Disk will use the appropriate sysparm (ARC0NAME, ARCCNAME, MERPnNAME or MERCnNAME) to create the data set name.

Assigning an Expiration Date to the Archive Data Set

By default, SAMS:Disk assigns the "never expire" date of 99365 to each of the archive volumes. This guarantees that an archive volume is never expired before all of the data sets that it contains. Therefore your only concern is the retention period being assigned to each data set being archived. This is recommended if you are using the EDM interface of your Tape Management System.

When SAMS:Disk determines that all data sets on an archive volume have expired, it automatically expires the volume as well. You can, however, specify a different expiration date through dynamic allocation with sysparm DYNEXPDT, or in JCL through the LABEL= parameter. Append either Eyyddd or Rddddd, where E indicates a Julian date follows and R indicates a 5-digit retention period follows.

For additional information, turn to the topic *"Customizing the SAMS:Disk Tape Management Support"* on page 62 in the *Installation Guide*.

Archiving to Disk: Requirements and Recommendations

Traditionally, backup and archive copies of data sets have been directed to tape. SAMS:Disk also provides the option of writing backup and archive data sets to disk. Using disk devices rather than tape or cartridge units for archive data sets eliminates the need for an operator to mount a tape when the backup or archive task is performed or when restoring data. This is particularly attractive for users implementing the auto-restore capability of SAMS:Disk.

Note: When archiving to disk, you should consider activating software compression to save on DASD space. To activate software compression, refer to the sysparm description of DCDATACP, beginning on page 137 in this manual.

The following summary outlines the steps necessary to implement archival to disk. Where necessary, an expanded explanation is provided for some of the summary items.

1. Specify archival to disk by one of the following methods:

- If your ARCHIVE and DMS JCL procedures do not contain //ARCHIVE0 or //ARCHIVEC DD statements, specify sysparm ARC0TYPE with a value of DISK for the primary, and/or ARCCTYPE with a value of DISK for the copy, to allocate them dynamically.
- Supply JCL statements for the ddnames above that will cause allocation to any disk device. SAMS:Disk will deallocate and then allocate the device as needed to perform the archive. For example, the following DD statement activates archival to disk for the primary copy:

```
//ARCHIVE0 DD UNIT=F6425,VOL=SER=anyvol,DISP=SHR
```

2. Specify the base name for the disk archive data sets using the following sysparms. SAMS:Disk will append a unique name to the end of the base name, based on the current date and time.

```
ARCONAME - for the name of the primary
ARCCNAME - for the name of the copy
```

3. Map the names you gave above to proper diskpools (members in your parmlib data set) from which target volumes will be selected. This mapping is specified with entries in member POOLDEFS of the parmlib data set. The general form of the entries is:

```
'VOLF6425 DMS/'
```

4. Place target volumes in your defined pools by creating the parmlib member by the same name as your pool, in this case "VOLF6425", and inserting statements such as:

```
VOL=(vol001,vol002,vol/)
```

5. Specify an expiration date using sysparm DYNEXPDT, or by JCL. SAMS:Disk will assign the default expiration date of 99365 if none is specified.
6. Determine if the default values blocksize is appropriate. If not, override the default with the sysparm ARCDSKBZ.

7. Determine if the default manner of calculating primary and secondary space allocation is appropriate. If not, override the default by specifying sysparms:

```
SPACEPRImmm – megabytes to allocate for primary
SPACESECmmm – megabytes to allocate for secondary
```

8. Determine if default values for index maintenance, merge and rebuild functions are appropriate, and if not, specify sysparm to override the defaults.

Step 1 — Specifying Archival to Disk

SAMS:Disk is distributed with default sysparms that cause archive and backup copies of data sets to be written to tape, creating both a primary and a duplicate copy tape concurrently. You may tell SAMS:Disk to write archive and backup copies of archive data sets to disk instead. You may write both the primary and duplicate copies to disk, or assign them to different media. You may also "dummy out" the duplicate copy. Indicate your choice of archive media either through JCL or SAMS:Disk sysparms ARC0TYPE and ARCCTYPE.

Step 2 — Naming the Disk Archive Data Set

For further discussion of this step, see *"Naming the Archives"* on page 61 in this manual.

Step 3 — Map Data Set Names to Diskpools

Candidate volumes for archiving to disk must be provided via "diskpools", which reside as members in the parmlib data set. The name of the diskpool is simply the member name. SAMS:Disk determines which diskpool (member) to use in the same manner in which tapepool names are determined. That is, the POOLDEFS member of parmlib is used to associate (map) a data set name to a poolname. For example:

```
'ARCPOOL0 DMS.ARCHPRIM/'
'ARCPOOLC DMS.ARCHCOPY/'
'BKPPPOOL0 DMS.BKUPPRIM/'
'BKPPPOOLC DMS.BKUPCOPY/'
```

Step 4 — Place Target Volumes in Defined Pools

Next, designate your candidate volumes in diskpool=ARCPOOL0 by creating a member in the parmlib data set with that name and inserting entries. For example:

```
VOL=(VOL001,VOL002,.....,VOL011)
VOL=(VOLA/,VOLB/,.....,VOLZ/)
```

A maximum of 100 volumes or patterns may be entered, with one or more volumes allowed per statement. Do not try to continue the volume list on multiple lines (that is, continuation lines are not supported). Instead, enter multiple VOL= statements.

If both the primary and the duplicate copy are being directed to disk, each pool must contain volumes of the same device type(s) as the other pool; that is, if pool 1 has

both F6425s and 3350s, pool 2 must also have some F6425s and 3350s. This is required because a volume from pool 1 is selected first, and then pool 2 is searched for a volume of the same device type.

By default, SAMS:Disk automatically separates the copy from the primary; that is, if a volume appears in both pools, SAMS:Disk will not select the same volume for the copy as is used for the primary. (Obviously, this is to prevent the loss of both copies if the disk pack is destroyed, whether it be due to oxidation, head crashes, or other reasons.) If you wish to allow both copies to go to the same volume, specify sysparm ARCSEPC0 with a value of N.

Step 5 — Specifying an Expiration Date

For further information about this step, see *"Assigning an Expiration Date to the Archive Data Set"* on page 62 in this manual.

Step 6 — Specifying Blocksize

When archiving to disk, SAMS:Disk automatically adjusts the optimal blocksize value for the device type of the volume; that is, half track blocking on 3375s and F6425s, and full track blocking on other devices. You may override the defaults by specifying sysparm ARCDSKBZ. The blocksize selected for a disk device is also used for any tape or F6470 copy that is being created concurrently.

If you have more than one device type in your archive pool, use sysparm ARCDSKBZ to specify a blocksize appropriate for all devices; that is, one that provides optimal performance and space usage on all the devices.

Step 7 — Specifying Space Allocation

SAMS:Disk determines the amount of space to allocate (the amount of space it needs on a diskpool volume) in the following manner. It assumes that quite frequently the primary copy will be on disk and the duplicate will be on tape or a F6470 cartridge. As stated earlier, the target volume leads SAMS:Disk to the device type, which in turn dictates the blocksize to be used. This, coupled with the density and length of tape being used, allows an easy and very accurate calculation of the tape capacity. (Since the ends of F6470 tapes cannot be clipped off, the most common variable in the calculation has been removed!)

The optimal disk allocation is the exact same capacity as its backup medium, such that when one is filled and both must be closed, nothing is wasted on the other medium. To do this on an exact equivalence, however, would require very large amounts of free space to be available. Therefore SAMS:Disk selects 1/16th of that value as default primary and secondary space allocations. In many cases, an archive run will not need a full tape, and one or two extents will be more than enough space. However, when a large archive run is made and the diskpool volumes contain plenty of free space, SAMS:Disk will obtain as many as possible of the 16 extents before closing the archive data set. This maintains high tape usage as well. Any excess (unused) disk space is, of course, released immediately when the data set is closed.

SAMS:Disk also considers the size of the data sets being archived when determining how much space to allocate. If the first data set to be archived is larger than the default primary space, the primary space is reset to the size of the data set. If free space equal to this new primary value is not available but the default value is, SAMS:Disk will attempt to archive the input data set to a multivolume output data set. This is done only as a last resort.

Once the target archive data set is allocated and the first data set to be archived is copied to it, SAMS:Disk will continue to copy additional data sets (the second through nth) as long as there is sufficient space to hold them. To determine whether sufficient space is available, SAMS:Disk examines the unused space in the current extent, and checks to see if additional extents can be obtained. If there is not sufficient space, the current disk archive data set is closed and a new one is allocated (like swapping to a new tape).

If your analysis or practical experience indicates that the default method of calculating space allocations should be changed, specify sysparms SPACEPRI and SPACESEC with a three-digit number representing the number of megabytes of disk space to allocate for each.

When archiving to disk, you will normally want the archive data sets to be large, as restoring from a large data set is as fast -- or faster -- than restoring from several small data sets. SAMS:Disk provides options for balancing the need to conserve storage space with the need to keep overhead to a minimum. If you set the primary space value (sysparm SPACEPRI) to the size you want to use for staging, then set the secondary space value (sysparm SPACESEC) to 000, SAMS:Disk will create smaller archive data sets, but more of them.

Step 8 — IXMAINT, MERGE and REBUILD

To accommodate archival to disk, the following options are available within the functions listed below.

- **IXMAINT**

Parameter SCRATCH=YES/NO is available on the DSNDELETE, VOLDELETE, and PURGE commands. DSNDELETE and VOLDELETE default to SCRATCH=YES, which means to scratch and uncatalog the disk archive data set when the archvols record is being deleted. PURGE defaults to SCRATCH=NO, because it is frequently used just prior to running REBUILD. If a disk archive data set is scratched, there is nothing from which to rebuild. If you do not intend to follow PURGE with a REBUILD command, specify SCRATCH=YES. Otherwise you will be creating "disconnected" archive data sets on disk that will never be used or deleted by any other means.

- **MERGE**

The following parameters have been added.

TYPE=(DISK,TAPE,F6470) — specify any of the three types listed to restrict the merge to archvols of those types.

CREDIT=date — only those archvols created on or before the specified date will be merged forward.

DAYSOLD=dddd — an "alternative form" of the CREDIT parameter that limits the merge to those archvols that were created dddd or more days ago.

- **REBUILD**

Rebuild accepts the archive data set to be read as input from either tape or disk. The supplied JCL procedure requires that you provide the correct //ARCHIVES DD statement pointing at the tape or disk data set to be rebuilt. No input cards are required. If they are provided, only the expiration date parameter is used.

Using Archive Functions to Manage Active Data

Traditionally, most archived data sets have been truly "inactive" data, and a very small percentage of archived data sets were ever restored. As users begin to use archive functions to manage active as well as inactive data, the number of data sets being restored increases dramatically.

Conceptually, it is probably more accurate to think of these archived active data sets as just temporarily compressed versions, not truly archived versions. In general, users do not want to keep the compressed versions archived very long, if at all, after the data set has been restored, as this wastes disk space.

SAMS:Disk provides sysparms that allow you to balance the need to manage active data sets with the need to use disk space most efficiently. This is accomplished by creating one archive data set for each data set processed, then allowing the disk space to be reclaimed quickly after the data set is restored from its compressed archive copy.

Here is how this process works. Sysparm RESIXRPD allows you to determine whether to keep the archived version of a data set after it has been restored, and if so, for how many days, by resetting its expiration date. The default value is 99, indicating no change is to be made to the expiration date. Specifying a value of 1 will cause the DSNINDEX entry to be kept one day after the restore. A value of zero will cause it to be deleted immediately. Note that this affects only the DSNINDEX entry. An exception to this rule is that no change will be made to the expiration date, nor will the DSNINDEX record be deleted, if the NEWNAME, NOLOAD, MEMBER or AIXNAME parameters are specified on a RESTORE command.

Sysparm RESIXDLB may be used in conjunction with RESIXRPD to indicate if DSNINDEX records are also to be deleted (or the expiration date reset) whenever a backup copy is restored. A backup copy will not have the scratch flag turned on in its DSNINDEX record (the data set was not scratched at archive time). The default value N tells SAMS:Disk not to reset or delete backup DSNINDEX records; that is, only the restore of an archived data set will cause the index record to be deleted. A value of Y tells SAMS:Disk to process all DSNINDEX records alike; that is, the value set in RESIXRPD will also control processing of DSNINDEX records for backup copies.

Then sysparm ARCHPACK can be specified with a value of N during the archive run, telling SAMS:Disk that "data set packing" is not to be used. The default value Y tells SAMS:Disk to pack all of the data sets being archived into a single data set. This packing feature alone saves, on the average, one half track of space for every data set processed. When N is specified, packing is not used (that is, SAMS:Disk creates one archive data set for each data set processed), and disk space is wasted. However, the benefit is that the disk space can be reclaimed quickly after the data set is restored.

Note: This sysparm should be specified as N only for archiving or merging to disk. If used when archiving or merging to tapes or cartridges, each tape will contain only one data set!

As an alternative to turning off the archive packing function as just described, you should consider using the disk-to-disk ability of the merge function to reclaim space occupied by expired data sets in the disk archives. A job to merge only the disk archives will not require any operator intervention, and can therefore be scheduled to run on a regular basis. This implementation maintains the advantages of data set packing, keeps the number of ARCHVOLS entries to a much smaller number, and still reclaims the expired "dead" space.

Archiving the Archives is Not Allowed

Traditional concepts naturally create a desire to use retention control to create more backups of the archive data sets being placed on disk -- or to "archive the archives." If this were permitted, it could lead to mass confusion in trying to manage the archives. For example, you might need to restore the archive copy of the archives in order to restore an archived data set. And if the archive copy of the archives was given a different retention period than the original archives, then it may have expired, and hence you can't get the original archives back, unless maybe you took a backup copy that was given a different expiration date, etc., etc. The desire for backups is well placed, but the method is wrong. Instead, SAMS:Disk allows you to create a duplicate backup copy at the time of the original archive run. Then, if the primary copy is lost, the ARCHVOLS record for it is disabled, and the backup copy will be used. If you need additional backup copies, use the archive copy utility.

Note: Creating the duplicate backup copy during the original archive run also requires less processing. The input (user) data is read once and then written to two different output devices -- read, write, write. Using ARCHIVE or RETAIN to cre-

ate another copy means an additional read -- read and write for the first copy, then read and write it again, increasing I/O by one third.

For the above reasons, SAMS:Disk will not allow archive and retention control to process the archive data sets written to disk. They are automatically detected and bypassed. The Move/Copy function can be used to copy them, however, since it will not introduce confusion to the archive/restore process. Moving the archives to other disk volumes may indeed be desirable if you are installing new devices, or perhaps just changing volumes within the archive diskpools.

Archive Integrity and Security

Archive Tape Protection

You can set your security package to assign WRITE (UPDATE) authority to users who must run backup, archive or merge jobs to SAMS:Disk archive data sets on tape. You can then assign an authority of NONE to all other users. This allows only authorized users to read, modify or delete these archives through utilities outside of SAMS:Disk.

SAMS:Disk Restore processing will take care of its own access to the archives, if the SAMS:Disk sysparm TAPEPRTR is specified with the default Y. To determine the data set names of the SAMS:Disk archives, see the discussion under "*Naming the Archives*" on page 61 in this manual.

SAMS:Disk password-indicates each archive tape, but does not create any passwords for them, if you specify the SAMS:Disk sysparm TAPEPROT with a value of Y. Also, this password-indication will not allow users to read, modify or delete these archives through utilities outside of SAMS:Disk. This protection is selectable, and would supplement any protection provided by your security package.

Checks Against Overwriting Valid Tapes

When a new archive or backup tape is to be created, SAMS:Disk always checks the archive index to see if it is already listed as having valid archive data. This may happen under the following conditions:

- An operator mounts the wrong tape in response to a request to mount a scratch tape.
- You have placed tapes into a pool for the first time, but they have already been used.
- The status of the SAMS:Disk archive index has not been kept synchronized with your tape management control data set. Your tape management system may have indicated that the tape is an available

scratch, but SAMS:Disk maintenance jobs to indicate it as such have not yet been run.

SAMS:Disk will reject a tape that is mounted but found to be already in use (it still has valid data on it), and call for a new tape. Additionally, if short tapes are mounted and tapepools are being used, the next tape will be selected from the appropriate pool.

For non-DSEF systems there are two different cases:

- If tapepools are being used, the tapes are validated just as in DSEF systems.
- When tapepool support is not being used, however, SAMS:Disk does not know the volume serial of the tape until after the operator has mounted it in response to the "mount scratch tape" request. If SAMS:Disk sees that it is already in the archives at this point, data that is possibly very valuable is on the verge of being overwritten. In fact, since the tape label has already been written, the first data set on the tape may be unrecoverable. To prevent any further damage, SAMS:Disk issues appropriate messages for the tape that may need to be recovered, and then shuts down, keeping the remaining data sets on the tape intact.

Tape Status After Abends

The SAMS:Disk archive index is updated immediately after each disk data set has been successfully copied to a tape. With this implementation, every archive or backup tape created by SAMS:Disk is a good tape, even though a system failure or job cancel may have occurred.

Note that if you are using a tape management system, its default action may be the opposite, that is, it may return a tape in use at abend time to scratch status. Consult with your tape management personnel to either change this action for tapes created by SAMS:Disk or take other precautions to ensure that a partial archive or backup tape is not marked as a scratch tape, and therefor possibly written over by some other job.

Using the External Data Manager (EDM) interface of a tape management system with SAMS:Disk is highly recommended. It provides the best method of securing SAMS:Disk tapes, including those being created as a result of an abend.

If CA1 release 4.8 or higher is installed and the SAMS:Disk/CA1 EDM interface is used, a tape created by an abending SAMS:Disk job will be controlled by SAMS:Disk and released only when the data sets on the tape have expired. See *"Customizing the SAMS:Disk Tape Management Support"* on page 62 in the *Installation Guide*.

Note: Contact Customer Support for information about other EDM interfaces that may be available.

If CA1 release 4.7 or below is installed and the SAMS:Disk/CA1 direct interface is used, a CA1 exit (member SLI035 in the USERMODS library) is available for the proper treatment of SAMS:Disk abend tapes. See *TMSCTLEX* on page 290 for more details.

If the SAMS:Disk tapepool support is used, tapes are usually indicated to the tape management system as being assigned permanently to SAMS:Disk and physically separated from the general scratch pool available to non-SAMS:Disk jobs. Since SAMS:Disk knows that the partial tape was written, it will not be called from the pool until its expiration date has passed. See *"Tapepool Considerations"* on page 43 for more details.

Archive Disk Protection

You may set your security package to assign all users the authority of NONE to all SAMS:Disk archive data sets on disk. This will prevent utilities other than SAMS:Disk from reading, modifying or deleting these archives. SAMS:Disk will take care of its own access to the archives. To determine the data set names of the SAMS:Disk archives, see the discussion under *"Naming the Archives"* on page 61 in this manual.

Unless you set SAMS:Disk sysparm DISKPROT to a value of N, SAMS:Disk will also password-indicate each archive data set on disk, but without creating any passwords for these archive data sets on disk. This password-indication will also prevent utilities other than SAMS:Disk from reading, modifying or deleting these archives. SAMS:Disk will take care of its own access to the archives. This protection is selectable, and would supplement any protection from your security package.

Processing PDSs that Contain Anomalies

To maintain data integrity for partitioned data sets, SAMS:Disk performs extensive validity checking against the VTOC entry, the directory and the member data itself. In practice, several inconsistencies may exist at the same time, or a single error may produce multiple error symptoms. In some of these circumstances it may be impossible to distinguish actual causes from the one or more effects, or to determine a "proper" course of action if an attempt to proceed is to be made. PDS anomaly processing does not apply to PDSE in any SAMS:Disk function.

If anomalies are found that may affect processing, SAMS:Disk will (by default) issue appropriate diagnostic messages and bypass the data set, relying on the assumption that someone will review the messages and correct the data set. However, provisions have also been made to allow processing to continue for certain types of errors, as defined below.

For each specific PDS error that SAMS:Disk detects, a 3-digit code has been assigned. The first digit is used to group similar errors together into a common "class" (which is associated with a particular stage of SAMS:Disk processing), and the last two digits are the specific anomaly code. As each stage of SAMS:Disk processing is completed, the decision to proceed to the next stage is determined by an error threshold associated with each specific anomaly, and another threshold associated with the class as a whole.

The threshold value is a percentage from 000 to 100, which represents the number of items in error compared to the total number of items in the class. Class "0" items are directory blocks, so threshold values refer to the percent of directory blocks that contain errors. Class "1" and "2" items are the members, so threshold values refer to the percent of members that contain errors.

The default threshold of 000 for each anomaly code causes SAMS:Disk to bypass any PDS in which an error is detected. Specifying 100 (or any value up to 999) causes SAMS:Disk to "accept" all errors of that type and continue processing as defined for that error. Threshold percentages between 000 and 100 therefore represent a "degree of error" which you are willing to accept before causing the data set to be bypassed. Every bypassed data set should be examined closely to determine what corrective action is needed.

The first code in each class does not represent a specific type of error itself. Instead, it represents a summary or composite entry for the entire class of errors, for which you can provide a different threshold. For example, you might want to allow processing to continue if a PDS has I/O errors on 5 percent or fewer of its members (threshold 005 for code 201), and with no more than 2 percent of its members diagnosed as disconnected aliases (threshold 002 for code 207). If there is no overlap between these two error groups, 7 percent of the members could be in error. A threshold of 007 for summary code 200 would allow processing to continue for such a data set. A threshold of 005 for code 200 would allow your "worst case" for either code, but not for different members in the same data set.

If you set the summary threshold below any of the specific error thresholds, you effectively lower them to the same value. (A special case is that of leaving the summary threshold at 000, which nullifies all thresholds for its class.)

Specifying your Options for PDS Error Handling

For some specific error types, SAMS:Disk provides more than one processing option. You specify both the processing options and the error thresholds in a member of the parmlib data set. This member may be given any name, and is identified to SAMS:Disk by sysparm PDSANMLYmmmmmmmm.

To change the default PDS anomaly options, create a new member with a name of your choice, and copy SAMPPDSA into it. This sample anomaly table contains all

of the error codes and default options in the required format. It will look like the following:

*PARTITION DATA SET ANOMALY CODE AND OPTION SPECIFICATIONS *			
*ANOMALY	THRES-	PROCESS	ERROR
*CODE	HOLD	OPTION	DESCRIPTION

'000	000		DIRECTORY SUMMARY
'001	000		*DIRECTORY I/O ERROR
'002	000		*ILLEGAL DIRECTORY FORMAT
'003	000		DIRECTORY KEY 8X"FF" MISSING
'004	000		DIRECTORY BLOCK KEY SEQUENCE
'005	000		DIRECTORY KEY NOT IN BLOCK
'006	000		DIRECTORY MEMBER(S) SEQUENCE
'007	000		DIRECTORY BLOCK ACTIVE BYTES
'100	000		EXTENT VIOLATION SUMMARY
'101	000		MEMBER BEYOND EXTENTS
'102	000	0	MEMBER BEYOND LSTAR
'*		0/A	.exclude member
'*		1/A	.attempt to process member
'103	000		MEMBER IN DIRECTORY
'200	000		MEMBER SUMMARY
'201	000	0	MEMBER I/O ERROR
'*		0/R	.restore whatever data was saved
'*		1/R	.bypass the member
'202	000	0	MEMBER NOT FOUND
'*		0/R	.member restored with EOF only
'*		1/R	.member not restored
'203	000		DUPLICATE REAL MEMBER(S)
'204	000		NOTE LIST BLOCK, SIZE ERROR
'205	000		NOTE LIST UNRESOLVED TTRXs
'206	000		MEMBER UNRESOLVED TTRNs
'207	000	0	DISCONNECTED ALIAS
'*		0/R	.restore as disconnected
'*		1/R	.restore, but convert to real
'*		2/R	.delete on restore
'208	000	0	MEMBER RECORD GT BLKSIZE
'*		0/A,R	.display size of largest record
'*		1/A,R	.display largest size per member
'209	000		NOTE LIST NOT FOUND
'210	000		OVERLAPPING MEMBER DATA
'211	000		INCONSISTENT USER TTR COUNT
'*		0/R	.leave user TTR count unchanged
'*		1/R	.adjust user TTR count
'*		2/R	.set user TTR count to zero
'212	000		PREMATURE END OF MEMBER (EOE)

Figure 3-2. PDS Anomaly Table

PDS Anomaly Code Descriptions and Notes

For each anomaly code, the threshold must be raised above 000 if processing for the data set is to continue after encountering the associated error. For anomaly codes 102, 201, 202, 207 and 208, SAMS:Disk provides a choice of processing options as described by the comments beneath each. The /A, /R and /A,R indicate whether the option applies to archive, restore, or both. The digit for the selected option must be placed on the line with the anomaly code and threshold.

Anomalies 001 and 002 will always result in the data set being bypassed, regardless of the threshold value. The threshold value for anomaly 208 is always ignored (forced to a value of 100), since data integrity is maintained in all cases. Its processing option simply gives control over the number of messages that will potentially be issued.

An important consideration for recovery purposes is the possibility for SAMS:Disk to, by default, never backup a PDS because of anomalies. Therefore, any PDS anomalies discovered should be corrected in a timely fashion.

Password protection is in effect only when not overridden by RACF or other security systems.

Processing Defined for Each Anomaly

For each anomaly with a threshold above 000, SAMS:Disk will continue processing as defined below.

Table 3-9. PDS Anomaly Codes and Descriptions

Code	Processing Description
000	<p><i>Percentage of directory blocks with errors</i></p> <p>This code does not represent a specific error type, but allows a threshold to be specified representing an acceptable (to you) percentage of directory blocks that can be found to contain errors, regardless of the type of error, and still allow processing to continue.</p> <p>Specific error types and a threshold for each one are defined below. The processing that will take place for each error that is "accepted" is also defined.</p>
001	<p><i>Directory I/O error</i></p> <p>Logical processing for the data set is terminated whenever an I/O error occurs processing the directory, regardless of the threshold value.</p>

Code	Processing Description
002	<p data-bbox="639 254 932 296"><i>Illegal directory format</i></p> <p data-bbox="639 317 1464 485">A PDS directory is defined as containing one or more records with a key length of 8 bytes and data length of 256 bytes followed by an end-of-file record. It begins at TTR=0000.01, the first record in the data set, and ends within the first extent of the data set.</p> <p data-bbox="639 506 1464 611">Logical processing for the data set is terminated whenever the above conditions are not found, regardless of the threshold value.</p>
003	<p data-bbox="639 632 1062 674"><i>Directory key of 8X'FF' is missing</i></p> <p data-bbox="639 695 1464 831">The end of the active directory is signified by a directory block with a key length of 8 bytes with all bits on, 8X'FF'. A non-zero threshold allows processing to continue even though this key is not found.</p>
004	<p data-bbox="639 852 1078 894"><i>Directory block is out of sequence</i></p> <p data-bbox="639 915 1464 1041">PDS member names should be unique and in ascending sequence. The key of each directory block should be the value of the last member name in the block. Therefore, directory block keys should also be unique and in ascending sequence.</p> <p data-bbox="639 1062 1464 1146">If a non-zero threshold is specified, a directory block whose key is out of sequence will be skipped.</p>
005	<p data-bbox="639 1167 1273 1209"><i>Directory block key is not the high member name</i></p> <p data-bbox="639 1230 1464 1293">Each directory block key should be the value of the last member name in the block.</p> <p data-bbox="639 1314 1464 1419">If it is not but a non-zero threshold is specified, processing continues but member names greater than the block key are bypassed, effectively deleting them from the copy being taken.</p>
006	<p data-bbox="639 1440 1224 1482"><i>Directory member names are out of sequence</i></p> <p data-bbox="639 1503 1464 1604">Directory member names should be unique and in ascending sequence. If they are not but the threshold value is non-zero, duplicate and descending member names will be skipped.</p>

Code	Processing Description
007	<p><i>Directory block active byte count is wrong</i></p> <p>The directory block active byte count is located in the first two bytes of each directory block. Its value is the sum of the lengths of all member entries, plus two for the length of the active byte count field itself. For all active directory blocks, the valid range for this field is 14 to 256.</p> <p>As each entry is processed, a residual byte count is computed by subtracting the length of the entry from the active byte count. Processing of the entries in the block stops when the residual count goes to zero or becomes negative. If all is well, the count will be zero. If it is negative, the active byte count was wrong and processing will be terminated for the data set unless a non-zero threshold has been specified.</p>
100	<p><i>Summary of extent violations</i></p> <p>The TTR value of each member entry is tested to see that it points within the data set, within the used space, and not within the directory itself. Each error is logged against its specific code below, but all are logged against this summary entry as well. This permits a threshold value to be specified for this code that represents the percent of members in each data set that can have extent violations, regardless of type, before processing for the data set should be terminated.</p>
101	<p><i>Member is beyond the extents</i></p> <p>If a member's TTRC points outside the data set's allocated space, a non-zero threshold causes the member entry to be ignored (it will not appear in the copy being taken).</p>
102	<p><i>Member is beyond the LSTAR address</i></p> <p>If a member's TTRC points outside the data set's used space and a non-zero threshold is specified, a 0 in the option field causes the member entry to be ignored (it will not appear in the copy being taken). A 1 causes SAMS:Disk to attempt to process it, for which the results are unpredictable.</p>
103	<p><i>Member points within the directory itself</i></p> <p>PDS member data should start immediately after the end-of-file record for the directory. Any TTR that points into the directory itself is invalid. (One exception is permitted, which is that alias entries may have a TTR=0000.01; that is, they may point to the very first directory block.)</p> <p>If a non-zero threshold is specified, the invalid entries will be bypassed, effectively deleting them from the copy.</p>

Code	Processing Description
200	<p><i>Percentage of members with internal errors</i></p> <p>As the data for each real member (and its associated alias entries) is processed, one or more of the errors as described below may be detected. The error threshold for this code represents the total percent of members that can have internal errors of any type before processing for the data set will be bypassed.</p>
201	<p><i>Member I/O error</i></p> <p>If an I/O error occurs while processing member data and a non-zero threshold is specified, processing for the member will continue on the next track. At restore time, processing will occur in one of two ways. If a 0 is specified as the processing option, whatever member data was read and saved successfully, if any, will be restored. It should be noted that some records (and potentially all of them) may have been lost. If a 1 is specified, no attempt is made to restore the member.</p>
202	<p><i>Member not found</i></p> <p>Each directory entry specifies a TTR location where the first block of its member data should be found. This error is indicated when an attempt to read the specified block results in a "record not found" condition. (A special case in which this condition is not regarded as an error is described under the topic <i>"Anomalous Conditions Not Regarded as Errors"</i> on page 80 in this manual.)</p> <p>Two options are available when a non-zero threshold permits processing to continue. A 0 causes the member entry to be maintained, but as a null entry (no data, just an end of file). A 1 causes the member entry to be bypassed, effectively deleting it from the copy.</p>
203	<p><i>Duplicate real members detected</i></p> <p>Real members should have unique TTR values to identify their locations. In this case, two or more have been found to point to the identical area. An examination of the data is required to determine which are correct and which are in error.</p> <p>If a non-zero threshold permits execution to continue, the duplicate entries will be maintained.</p>

Code	Processing Description
204	<p data-bbox="545 262 899 294"><i>Note list size error detected</i></p> <p data-bbox="545 325 1377 514">A user TTRN in the directory entry identifies a note list block at location TTR if the N field has a value not equal to zero. This value also indicates the size of the note list block; that is, the data length is the quantity N times four. An error is detected when the block at that location is read but does not have the proper length.</p> <p data-bbox="545 546 1377 667">If a non-zero threshold permits processing to continue, the block in question is treated as a standard data block; that is, no attempt will be made to treat it as a note list block and resolve internal TTRXs.</p>
205	<p data-bbox="545 699 899 730"><i>Unresolved note list TTRXs</i></p> <p data-bbox="545 762 1377 951">If a note list block of the proper length is found, it should contain TTRXs pointing internal to the member data itself. SAMS:Disk validates these TTR values such that they can be updated when the member is moved. SAMS:Disk also permits TTR values less than or equal to 0000.01 to remain unchanged. An error is detected when the TTR fails to satisfy the above tests.</p> <p data-bbox="545 982 1377 1077">If a non-zero threshold is specified, these invalid TTRs are left unresolved when the member is moved, which may lead to unpredictable results in the future.</p>
206	<p data-bbox="545 1104 922 1136"><i>User TTRN can't be resolved</i></p> <p data-bbox="545 1167 1377 1388">A directory entry for a member may contain up to three user TTRs that point to records within the associated member data. Unlike the member's TTRC, these TTRs must always point to the exact location of a data record. (SAMS:Disk also permits TTR values less than or equal to 0000.01 to remain unchanged.) An error is detected when the TTR fails to satisfy the above tests.</p> <p data-bbox="545 1419 1377 1514">If a non-zero threshold is specified, these invalid TTRs are left unresolved when the member is moved, which may lead to unpredictable results in the future.</p>

Code	Processing Description
207	<p data-bbox="643 254 1008 285"><i>Disconnected alias detected</i></p> <p data-bbox="643 317 1469 453">An alias entry has a TTRC that points within its associated real member. (For load modules, the TTRC of an alias entry is the same as the real member's, but the entry point addresses may differ.)</p> <p data-bbox="643 474 1469 705">An error is detected when the TTR value of an alias entry fails to point internal to any real member data. (These are commonly caused by improper specification of alias names when using the linkage editor or JSECOPY to process a member that has aliases. The "old" alias entries were left unresolved, and therefore point to what probably should be considered "down-level" copies of the original member data.)</p> <p data-bbox="643 726 1469 926">If a non-zero threshold permits processing to continue, there are three options that may be taken, all of which apply at the time the member is restored. A 0 causes the entry to be maintained as a disconnected alias. A 1 will cause it to be converted to a real member. And a 2 will cause the member to be omitted, effectively deleting it from the new data set.</p>
208	<p data-bbox="643 947 1179 978"><i>Member record greater than the blocksize</i></p> <p data-bbox="643 1010 1469 1199">The F1-DSCB specifies the maximum size of all member records in the PDS. An error is detected when a data block is read whose length exceeds the stated value. SAMS:Disk maintains data integrity by processing the entire data block, regardless of its length. Other applications may, however, truncate these records, which will cause unpredictable results.</p> <p data-bbox="643 1230 1469 1482">For the above reasons, SAMS:Disk always continues processing for this error type, regardless of the threshold value. A processing option of 0 causes a warning message to be issued, indicating the blocksize of the largest block for the data set in error. Option 1 will provide this information, plus messages indicating the blocksize for information, plus messages indicating the blocksize for the largest block for each individual member in error.</p>
209	<p data-bbox="643 1503 951 1535"><i>Note list block not found</i></p> <p data-bbox="643 1566 1469 1703">A user TTRN in the directory entry identifies a note list block at location TTR if the N field has a value not equal to zero. This error is detected when an attempt to read the block at that location results in a "record not found".</p> <p data-bbox="643 1724 1469 1829">If a non-zero threshold is specified, processing will continue but the invalid user TTRN will be left unresolved, which may lead to unpredictable results in the future.</p>

Code	Processing Description
210	<p><i>Overlapping members</i></p> <p>When processing members in their ascending TTR sequence, the end of file for each member should occur prior to the TTRC for the beginning of any subsequent members. An error is detected when this is not the case. The member that has just been read is considered good. All subsequent members whose beginning TTRs have been overlapped are considered invalid.</p> <p>If a non-zero threshold is specified, processing will continue but the invalid members will not be restored.</p>
211	<p><i>Inconsistent user TTR count</i></p> <p>More user TTRs are indicated than can be contained within the user data field of this member. When the member is read, all user TTRs outside of the range of the user data fields are ignored. When the member is restored, three processing options are available. If processing option 0 is specified, the directory entry for the member is written back unchanged. If option 1 is specified, the directory entry for the member is restored with the user TTR count adjusted to reflect only the TTR count that can exist in that member's user data field length. If option 2 is specified, the user TTR count is set to zero when the directory entry is restored.</p>
212	<p><i>End of extents encountered prior to end of file</i></p> <p>The end of every PDS member is indicated by an end of file record. This error is detected when the end of the data set is reached before encountering the end of file.</p> <p>If a non-zero threshold is specified, the member will be processed as if the end of file had been reached instead of the end of the data set; that is, the member is assumed to consist of the data processed up to that point.</p>

Anomalous Conditions Not Regarded as Errors

The following anomalous conditions can be found in PDSs, but their interpretation and processing is common among Fujitsu and many vendors, and are therefore not regarded as errors.

1. Member TTRC not found on track

A member's TTRC should point to the relative location within the data set of the member's first record. Occasionally this TTRC will point incorrectly to the end of a track (to a record that does not exist) instead of pointing to the beginning of the next track where it was actually written.

This is usually caused by a PDS update program thinking it can start a member at the end of a track but then discovering that the data block is too large to fit in the remaining space on the track. It therefore writes on the next track, but "forgets" to go back and update its starting location, resulting in the TTRC value being technically in error.

Knowing how this condition arises, read routines will automatically switch to the next track and return the first record, without indicating an error condition.

2. Alias entries with TTR=0000.01

Although technically invalid for a standard PDS, this TTR value can be left unchanged without any integrity exposure, and is therefore not regarded as an error. (Even when moved to a different device type and location, the TTR value never needs to be resolved -- record one will still be record one. Even if the size of the directory is reduced, its minimum size is at least the one block.)

Other vendor products (for example, STROBE) are known to create and use these entries intentionally.

Error Messages and Associated Processing for Anomalies

When an anomaly is detected and a message issued, the error will be identified by its associated code and an appended letter. An appended A indicates that the error was detected at archive time, reading the input data set. An R indicates that the error was detected at restore time in the target PDS. An F indicates that the error was in the data contained on the archive medium itself, which means that the errors were allowed previously when the data set was being archived.

When anomalies are detected but your threshold options have allowed the data set to be archived anyway, a "logical error flag" in the index record is set for the data set and "logs" the errors with the archived data set. This permits restore processing to inform you of those facts whenever you restore a copy that contained errors.

PFD Custom Reports

The following discussion assumes that the user has a working knowledge of the PFD online reporting feature of SAMS:Disk. You should be familiar with the *PFD* section, which begins on page 421 in the *User's Guide*, covering the online report facility, and also have had hands-on experience with the product. Although the online reporting facility of SAMS:Disk provides a lot of flexibility in defining user reports, it doesn't allow much tailoring as far as how individual fields are printed (column headings, numeric editing patterns, etc.). You can tailor these values on an installation-wide basis, however, by modifying the data dictionary in parmlib.

Each dictionary element requires two text lines in the parmlib member. These follow the standard rules for continuing information in a SAMS:Disk parmlib member. The starting column number listed is relative to the first character past the quote. The fields that should not be modified are so documented. No fields should be deleted from this member -- there are other options that allow the suppression of fields from selection lists.

Table 3-10. PFD Custom Report Field Names

Columns	Name of Field	Description
1-8	FIELD NAME	This is the 8-character field name that appears on the ISPF panels. This field should not be changed unless it is done prior to any reports being defined. If you wish to use a different field name
9-16	SOURCE FIELD IN FMTxFLDS	This is the name of the source field in the lower level dictionary. It should never be modified.
17-18	WIDTH OF PRINT DATA	This is a 2-digit number that tells the report writer how wide the field is on the output report. The heading that is specified may cause the actual width of the field to be enlarged on the report.
19-32	DATA CONVERSION METHOD	This field tells the report writer how to interpret the source field to get it into a printable format. There are several techniques that are used. The first character gives the method and the next 13 characters define modifiers where applicable. The conversion methods are:
		* No data conversion required

Columns	Name of Field	Description		
		M	A special module is required to convert the data into its proper format. Details covered later.	
		N	The field is to be printed with the numeric pattern specified (such as ZZ,ZZ9). The edit pattern should be the same length as the field width specified.	
		X	Print the field as a hexadecimal value.	
		G	Gregorian date. The format will be influenced by the width of the field. For example:	
			Width	Format
			6	MMDDYY
			7	DDMMMYY — where MMM is alphabetic (i.e., 12JAN99)
			8	MM/DD/YY
			9	DDMMMYYYY
			10	MM/DD/YYYY
			12	MMM DD, YYYY
		J	Julian date format. The format will be influenced by the width of the field. For example:	
			Width	Format
			5	YYDDD
			6	YY.DDD
7	YYYYDDD			
8	YYYY.DDD			
12	MMM DD, YYYY			

Columns	Name of Field	Description	
		E	European format. The format will be influenced by the width of the field. For example:
		Width	Format
		8	DD/MM/YY
		10	DD/MM/YYYY
33-40	SORT CONVERSION MODULE	If the source field will not sort properly in its natural state, a special conversion module is called to convert the data as needed. If no module is specified, the sort data is extracted directly from the source field.	
41-42	SORT DATA WIDTH	Used only when a sort module is specified. It gives the length of the converted sort data. This field should not be changed for established fields because SAMS:Disk-supplied routines require the length specified.	
43	PRINT JUSTIFICATION	How should the printed text under the column heading be justified? Choices are:	
		L	left justified
		R	right justified
		C	centered
		A	as is
		Can be changed to suit your tastes. If "A" is used for numeric fields, the editing pattern specified will determine where the field gets printed (Hint: to make the numbers line up on the right column, make the editing pattern as wide as the heading).	
44-45	SUBGROUP OCCURRENC	Not currently used.	

Columns	Name of Field	Description
46	FLAG - DISPLAY ON REPORT SCREEN?	Should the field be displayed on the field selection panel where the user selects the fields to appear on the report? (Y/yes N/no) You may not want the user to have access to all fields (such as RACF flags, catalog flags, etc.) Change this flag to an N and the user will no longer see it on the selection panel, and will therefore not be able to print it.
47	DISPLAY ON SORT SELECT SCREEN?	This field governs which fields get displayed on the ISPF selection panel for sort fields. A=always
48	DISPLAY ON TOTALS SCREEN?	This field governs which fields get displayed on the ISPF selection panels for sub and grand total fields. A=always, N=never, S=only if field was selected to be reported on. If you change an N flag to A or S, be sure that the source field is defined as numeric.
49-68/1-10	COLUMN HEADING FOR REPORT	The heading to use over the column where the value is printed on the report. The heading starts on the first definition line and continues on the next line. Up to 30 characters may be specified. The special character "]" indicates the heading should be split to another print line. There can be up to two of the "]" characters in any given title. Leading and trailing blanks are honored and help to govern the width used for printing the heading. Hint: If you want the headings to appear on the report in a language other than English, you can change them here and all reports will automatically reflect the new headings when they are run the next time.
11-52	FIELD DESCRIPTION	A free-form description of the field. This description appears on all ISPF panels where the field is referenced. Again, these descriptions can be changed to a language other than English for the convenience of your users. Any change implemented here will take effect for all users immediately. The description does not appear on any reports that are printed by the system.

Coding PFD Report Data Conversion Modules

The following documentation provides the minimal guidelines required to code a data conversion routine. Because of the narrow appeal of this type of facility, it will not be dealt with in an exhaustive manner.

There may be occasions when the standard methods of data conversion for printing purposes will not cover a user's requirement. The module exit facility allows the user to code special conversion routines and include them in the data dictionary. There are a few exits that are supplied with the SAMS:Disk system because of special editing requirements. These include the RACF indicator flags, catalog locate routine, and such fields as DSORG, RECFM, etc.

The module coded in the dictionary will be passed the following parameters when called upon to format a field for output:

Table 3-11. Parameters Passed to Dictionary Module

Parameter Description	Type	Length
field name (from dictionary)	INPUT	CL8
source field's length	INPUT	XL1
source data	INPUT	variable
converted data's expected width	INPUT	XL1
converted printable data	OUTPUT	variable

The field name is the source field name that is coded in the dictionary definition. This parameter can usually be ignored unless the same conversion routine is called to convert more than one field name.

The source field's length is extracted directly from the dictionary entry. This is used to tell the program how wide the source data (passed as parameter 3) is. The source data is the raw data that is found in the F1 or F4 DSCB.

The final two parameters govern the width and content of the formatted print data. The fourth parameter is used to tell the called program how wide the print area is for this field. The called program under no circumstances should return data that exceeds the length specified. If you need a larger print area, change the value in the dictionary to reflect the required size. Please note that the numeric fields passed are one-byte unsigned binary numbers.

Suppose you need the contents of more than one source field in the DSCB to determine what to print. This was a requirement of a couple of the standard print fields (for example, to print the catalog flag, both the data set name and volume were required to issue a catalog locate). The source data being passed actually resides in a contiguous area of storage that contains both the DSCB and appendage areas.

What happens in the call to the conversion module is that the offset to this field is calculated and then the called program only sees from the source field on in this storage area. So the way to allow addressing to all parts of this area is to specify the field data set name as the source field (since it is the first field in the F1-DSCB). Then you can use the following mapping macros to provide addressability to the entire area: ADSDMDS1, F1APPEND, and S1APPEND.

Coding PFD Report Sort Conversion Modules

The following documentation provides the minimal guidelines required to code a sort conversion routine. Because of the narrow appeal of this type of facility, it will not be dealt with in an exhaustive manner.

The method used to provide a user-written sort conversion routine is very similar to that used for printing purposes. The biggest difference is in the parameter list to the conversion module; there is no parameter describing the length of the source field for the sort. The other notable difference is that the output of the conversion routine does not get printed anywhere.

The module coded in the dictionary will be passed the following parameters when called upon to format a field for sorting:

Table 3-12. Parameter Passed to Module for Sorting

Parameter description	Type	Length
field name (from dictionary)	INPUT	CL8
source data	INPUT	variable
converted data's expected width	INPUT	XL1
converted sort data	OUTPUT	variable

The field name is the source field name that is coded in the dictionary definition. This parameter can usually be ignored unless the same conversion routine is called to convert more than one field name.

The final two parameters govern the width and content of the formatted sort data. The third parameter is used to tell the called program how wide the sort area is for this field. The called program under no circumstances should return data that exceeds the length specified. If you need a larger sort area, change the value in the dictionary to reflect the required size. Please note that the numeric fields passed are one-byte unsigned binary numbers.

User-Specified Condition Codes

SAMS:Disk assigns a step completion condition code to every step. The code accompanies a message and indicates the circumstances under which the step finished. In most cases, this SAMS:Disk-generated condition code will be the best value for your installation. However, you can override these generated condition codes if your installation has special requirements. This override allows you to select the code that accompanies a message upon the completion of a step. These user-specified condition codes are optional.

To specify user condition codes, you must create a member in PARMLIB called CCSET. This member is not provided with the system, although a member called SAMPCSET is provided as a sample, and can be copied to create your initial CCSET. User condition codes are then specified by an entry in this member.

When using this option, several technical points must be kept in mind with respect to user condition code processing.

1. Once an N (no-override) is encountered during processing, SAMS:Disk will determine, set and assign condition codes for all the message number entries that follow without regard for the overrides specified in member CCSET.
2. Whenever a message is issued during step processing, the highest condition code will be set without regard to its origin.
3. Before SAMS:Disk returns control to the operating system, a check of the SAMS:Disk-generated return code is made. If the SAMS:Disk return code is odd and the user condition code option is in effect, then the user override condition code will be incremented by one. For example, if the user specified an override condition code of 4 and the SAMS:Disk generated condition code was found to be odd, then the user override condition code of 4 would be incremented to 5. This is needed to produce sorted report listings.
4. No duplicate message number entries will be allowed in member CCSET. If duplicates are detected at table initialization, the program will abend with a user 200.

Each CCSET entry is composed of a message number, a state flag (O=override or N=no-override), and one of five possible condition codes. The message number is the first four positions followed by a blank. The next two positions consist of the state flag and a condition code respectively.

To add, change or delete entries in the CCSET member, use your online editor or other conventional means for updating a PDS member in the following manner:

1. Create a member named CCSET in the parmlib data set by copying and renaming the sample member SAMPCSET.

- Specify the message display control entry in CCSET. This must be the FIRST entry in CCSET and is required. This entry controls the display of the message which notifies users that the user condition code option is in effect. It is recommended that you try the default first which displays the message in BATCH and TSS. The format for this entry is:

Table 3-13. Format of \$FLGXyz

Field	Meaning		
\$FLG	\$FLG		
X	BLANK		
y	BATCH FLAG	B	display message under BATCH
		BLANK	do not display message
z	TSS FLAG	T	display message under TSS
		BLANK	do not display message

In order to suppress the display of the notify message under BATCH or TSS processing, simply blank out the respective flag. If you wish to suppress the display of the notify message under BATCH and TSS processing, blank out both flags.

- Specify the message number, state flag and condition code enclosed by apostrophes in the CCSET member. An example could be:

' 3179 OE '

The format for message number, state code and condition code is:

Table 3-14. Format of MMMMBsc

Field	Meaning
MMMM	message number
B	blank
s	state flag
c	condition code

The valid entries for these are:

Table 3-15. Valid entries for MMMMBsc

Field	Possible Values	
MMMM	any valid four-digit SAMS:Disk message number	
B	must be blank	
s	O	override
	N	no-override
c	G	0, good, successful completion
	I	4, informational message
	R	8, resource error
	E	12, error occurred during SAMS:Disk processing
	C	16, command error, DSCL command error detected

DASD Threshold Manager

Overview

The SAMS:Disk DASD Threshold Manager is a program that can help you maintain an acceptable level of freespace on DASD volumes or within a pool of DASD volumes. This program enhances the usefulness of SAMS:Disk without reducing or altering its functionality.

Most implicit SAMS:Disk storage management tasks scan VTOCS or catalogs to determine what data sets will be archived or moved, based on data set attributes. Data sets are archived or moved with the assumption that this action will achieve desirable storage management objectives.

The Threshold Manager begins with the aim of achieving a stated storage management objective: to maintain a specified percentage of freespace balance across a volume or a pool of volumes. Instead of examining data set attributes, the program examines volume (or volume pool) freespace. If the data on the volume increases beyond a specified threshold, causing the freespace to decrease below the desirable percentage, the program determines what data sets could be archived or moved to achieve the balance objective across the volume or pool.

Based on user-specified criteria, the program then generates explicit DSCL Archive or Move Migrate commands for input into SAMS:Disk. It does not invoke SAMS:Disk to perform the Archives and Moves directly, but acts as a front-end to SAMS:Disk.

The Threshold Manager determines the least amount of data to archive or move in order to achieve the balance. If you are processing a pool of volumes, the program will archive or move the least required amount of data. The remaining data sets will be distributed within the pool. Each volume will finish with an optimum balance of data and freespace.

The Threshold Manager procedure (THRSHMGR) dynamically allocates two sequential data sets for the DSCL Archive and Move commands it generates. These sequential data sets are specified on the //ARCHOUT and //MIGROUT DD cards in the procedure.

These files will become the SYSIN data sets for the DMS procedure. They will have an LRECL of 80 and will be large enough to handle three records for each complete Archive or Move command generated.

The Threshold Manager is driven by keywords and parameters supplied through the execution of the THRSHMGR procedure and the SYSIN DD statement. The keywords and parameters define the scope of the processing and the criteria used to determine the DSCL commands generated.

The Threshold Manager keywords, parameters and subparameters used in the job control statements use a different syntax than standard SAMS:Disk processing. This syntax is as follows:

Threshold Manager Control Statement Syntax

- A line is considered to begin at the first non-blank character.
- The line will be considered a comment if it begins with an asterisk or contains only blanks through column 72.
- A line will be scanned from column 1 through column 72 only.
- After the beginning character, each line is terminated either by a blank character or column 72.
- A line is continued if and only if its end character is a comma.
- A line and all of its continuation lines will be considered a statement. Only a keyword can begin a statement.
- Keywords and parameters can be abbreviated to their smallest unique leading characters.

- Keywords, parameters or values may not contain any blank characters. The only exception to this is when the USEREXIT parameter is present, its null value requests that an KDJBR14 lookalike be used.
- Any list of parameters or parameter values must be separated by commas and enclosed in parentheses. Parentheses are not required when using a single parameter or value.
- Keywords and parameters must be delimited with an equal sign which must be followed with its appropriate value or list of values.

Table 3-16. Threshold Manager Syntax Error Messages

Error Number	Message
100	'CONTINUATION EXPECTED'
101	'KEYWORD DELIMITER "=" NOT FOUND'
102	'MISSING KEYWORD'
103	'INVALID NUMERIC FIELD'
104	'MISSING VALUE'
105	'DUPLICATE POOLNAME'
106	'UNPAIRED PARENTHESIS'
107	'INVALID DSN'
108	'INVALID MEMBER NAME'
109	'DYNAMIC ALLOCATION FAILURE'
110	'INVALID PARAMETER FOR KEYWORD - keyword-'
111	'INVALID DATE'
112	'DSN PARAMETER IS REQUIRED'
113	'MISSING VALUE FOR PARAMETER'
114	'"/" CAN ONLY BE LAST CHAR OF A DSN PATTERN'
115	'DSN NAME EXCEEDS 44 CHARS'
116	'INVALID DSN VALUE'
117	'CONTINUATION NOT ALLOWED'
118	'INVALID KEYWORD'

Error Number	Message
119	'"YES" OR "NO" EXPECTED'
120	'DUPLICATE OR MUTUALLY EXCLUSIVE KEYWORD'
121	'DSNAME REQUIRED'
122	'MULTIPLE VALUES NOT ALLOWED'
123	'LINE CANNOT TERMINATE WITH "("'
124	'"GLOBAL" CANNOT FOLLOW A "POOLDEF"
125	'DDNAME STATEMENT DOES NOT EXIST'
126	'DDNAME MUST BE 1 TO 8 CHARS LONG'
127	'DUPLICATE OR CONFLICTING KEYWORDS'
128	'KEYWORD TRUNCATION IS NOT UNIQUE'
129	'INVALID CONTINUATION OF KEYWORD LINE'
130	'POOLNAME MUST BE 1 TO 8 CHARS LONG'
131	'THRESHOLD VALUE MUST BE IN THE RANGE 1-99'
132	'IS PRECEDED BY VOLSER DEFINITION'
133	'EXIT NAME MUST BE FROM 1 TO 8 CHAR'
134	'EXIT MODULE -modname- FAILED TO LOAD'
135	'INVALID CHANNEL ADDRESS VALUE'
136	'INVALID DEVICE'
137	'INVALID MOUNT STATUS'
138	'INVALID VOLSER PATTERN'
139	'INVALID SUBPARAMETER FOR PARAMETER - parmname'
140	'EXPDT MUST BE AFTER TODAY'
141	'LAST REFERENCE MUST BE BEFORE TODAY'
142	'NEITHER "MIGROUT" NOR "ARCHOUT" DD IS PRESENT'

Error Number	Message
143	'MIGRATE=YES BUT "MIGROUT" DD IS NOT PRESENT'
144	'ARCHIVE=YES BUT "ARCHOUT" DD IS NOT PRESENT'

Threshold Manager Control Hierarchy

Below is a summary of the keywords, parameters and subparameter hierarchy, and their default values. This is followed by sample JCL to execute the program. A more complete description of the keywords, parameters and subparameters is provided following the JCL.

```

GLOBAL=parameter/parameter list
    ARCHIVE=y/n
    EXPDT=+180/date format
    LASTREF=-7/date format
    VSAM=n/y
    ISAM=n/y
    THRESHOLD=80/1-99
    RECATALOG=y/n
    DELETE=n/y
    MIGRATE=y/n
    USEREXIT=user exit module name
EXCLUDE=DSN=dsn/dsn list/dsn pattern/dsn pattern list
INCLUDE=parameter
    DDNAME=ddname/ddname list
    or
    FILENAME=file name/file name list
MULTIVOL=n/y
POOLDEF=parameter/parameter list
    ARCHIVE=y/n
    EXPDT=date
    LASTREF=date
    VSAM=n/y
    ISAM=n/y
    THRESHOLD=80/1-99
    RECATALOG=y/n
    DELETE=n/y
    MIGRATE=y/n
    USEREXIT=user exit module name
    NAME=pool name

```

CHANADDR=channel address/channel address list
 DEVICE=device type/device type list
 MOUNT=pu/pr/s (mount status - public, private, storage)
 VOLSER=volser number/volume pattern/threshold values

JCL - Threshold Manager

The following figure illustrates the JCL required for a successful execution of the Threshold Manager:

```
//S2          EXEC  THRESHMGR
//S2.SYSIN DD  *
GLOBAL=(ARCHIVE=,MIGRATE=,VSAM=,ISAM=,EXPDT=,LASTREF=,
RECATALOG=,DELETE=,THRESHOLD=,USEREXIT=)
EXCLUDE=(DSN=a.b.c/,
          DSN=d.e.f/)
INCLUDE=DDNAME= OR INCLUDE=FILENAME=
MULTIVOL=
POOLDEF=(threshold=,name=,vol=(,),chanaddr=,device=,mount=,archive=,migrate=,
          recatalog=,delete=,vsam=,isam=,expdt=,lastref=,userexit=)
```

Figure 3-3. Threshold Manager JCL

Keyword, Parameter, and Subparameter Definitions

The Threshold Manager procedure uses five keywords. Each control statement must begin with a keyword. Although none of the keywords are required, at least one keyword must be supplied.

Each keyword is followed by a parameter, parameter list or value, as follows:

GLOBAL=

The GLOBAL= keyword is used to specify a series of parameters that set global default values. These default values will be honored unless they are overridden by a different value supplied on the POOLDEF= keyword. The GLOBAL= keyword may be specified only once, and must precede any POOLDEF keywords.

ARCHIVE= The default value of this parameter is NO, causing the program to bypass generation of Archive commands. A value of YES causes the program to generate DSCL Archive commands.

MIGRATE= The default value of this parameter is YES, causing the program to generate DSCL MOVE commands. Specify NO to bypass generating MOVE commands.

EXPDT= Specify a retention period or date that will be applied to the generated Archive commands. If this parameter is not specified, a default retention period value of +180 will be used.

Valid retention period format is "+nnnn", indicating today plus nnnn days. The date calculated from the retention period cannot exceed the year 2155.

Valid date formats are:

ccYYDDD	
ccYY.DDD	
ccYY/MM/DD	cc= Century (optional)
SYDDDD	S= Special Julian Dates
SYD.DDD	
SYDDDDDD	
SYDDDD.DDD	

LASTREF= Specify a time period or a date to cause data sets to become candidates for Archive that have not been referenced during this time or since this date. To specify a time period, use a negative number to indicate the number of days in the past.

If this parameter is not specified, the default time period value of -7 is used, indicating the data set has not been referenced for seven days.

If you specify a date, it must be in the past. Valid date formats are:

ccYYDDD	
ccYY.DDD	
ccYY/MM/DD	cc= Century (optional)

RECATALOG= The default value of YES causes archived data sets to be recataloged to the SAMS:Disk pseudo-volume. If you specify a value of NO, no recatalog action is taken at archive time. This parameter is mutually exclusive with DELETE. For related information, please turn to the *DISP=* parameter description on page 200 of the *User's Guide*.

DELETE= The default value of NO prevents any catalog action or scratch from occurring during archive time. If you specify a value of YES, archived data sets will be scratched and uncataloged. This parameter is mutually exclusive with RECATALOG. For related information, please turn to the *DISP=* parameter description on page 200 of the *User's Guide*.

THRESHOLD= The default value of this parameter is 80, which indicates that the desired freespace threshold is 80 percent (that is, 80 percent of the volume's tracks are allocated). You may specify any value from 1 to 99.

VSAM= The default value of this parameter is NO, which excludes EDF VSAM clusters from being candidates for Archive or Move processing. If you specify a value of YES, EDF VSAM clusters may be considered candidates for Archive.

ISAM= The default value of this parameter is NO, which excludes ISAM data sets from being candidates for Archive or Move processing. If you specify a value of YES, ISAM clusters may be considered candidates for both Archive and Move.

USEREXIT= Specify the name of a user exit module that exists and can be programmatically loaded. The default value is null and a routine equivalent to KDJBR14 will be linked as the user exit.

EXCLUDE=

The EXCLUDE= keyword is used to exempt specified data sets from processing.

The EXCLUDE= keyword must precede any POOLDEF= keywords.

DSN= The DSN= parameter is the only valid parameter for the EXCLUDE= keyword. It may not be truncated. Specify for this parameter a data set name or pattern or a list of data set names or patterns that are to be exempted from processing. Some examples of valid data set patterns are:

prefix/	exclude all data sets that begin with "prefix"
*.prod/	exclude production data sets
!abcd!	exclude data sets that end with "abcd".

You may supply one DSN= parameter on an EXCLUDE= keyword, or a list of DSN= parameters.

INCLUDE=

The INCLUDE= keyword is used to insert a sequential file or DD statement into the SYSIN. The file to be inserted will be substituted for the INCLUDE= statement, and must have a logical record length of 80. The include file's syntax must obey all syntax rules, as it will be interpreted in exactly the same manner as the instream control statements. Any number of INCLUDE= statements may be used or nested.

DDNAME= This parameter is used with the INCLUDE= keyword. Specify the DDNAME of the DD statement to be inserted into the SYSIN.

FILENAME= This parameter is used with the INCLUDE= keyword. Specify the data set name for a sequential file, or the PDS(member) name of the file to be inserted into the SYSIN.

MULTIVOL=

The default value of this keyword is NO, which excludes multivolume data sets from being considered as candidates for Archive processing.

If this keyword is specified with a value of YES, only those multivolume data sets with all of their volumes defined in the specified pool will be considered as candidates for Archive processing. If a multivolume data set is contained in part on a volume not defined within the specified pool, the data set will not be considered a candidate.

POOLDEF=

The POOLDEF= keyword is used to define a pool of volumes, limit the Threshold Manager's processing to volumes within the pool (or pools if multiple POOLDEF= keywords are supplied), and to indicate the processing criteria to use in selecting data sets on volumes in the pool to Archive or Move. By specifying a VOLSER= parameter, you may also define subpools within a pool of volumes and specify a THRESHOLD= subparameter to set a different threshold limit for volumes in the subpool. If this keyword is omitted, all volumes defined in the system's UCB table are candidates for processing.

The POOLDEF=keyword must be specified following any GLOBAL= or MULTIVOL=keywords. It can be repeated as necessary to define multiple pools.

Volumes will be selected and assigned to pools (and subpools, if a VOLSER= parameter is specified) in the order of definition. Once an online volume is assigned, it will be flagged so that it will not be reselected.

The 10 parameters defined above for the GLOBAL= keyword may be specified on the POOLDEF= keyword. If specified, they will override the global parameters for volumes in this pool. In addition to the 10 global parameters, six other parameters may be specified for the POOLDEF= keyword.

NAME= Specify a 1- to 8-character name to be assigned to the defined pool. If not supplied, a unique default name will be assigned to the pool.

CHANADDR= This parameter restricts the volumes that can be assigned to this pool by supplying valid volume channel addresses. If this parameter is not supplied, then all channel addresses are candidates for selection.

Specify one or a list of channel addresses, as defined in the UCB table. Each entry must be four characters in length. Each character must be a valid hexadecimal character (0-F) or the wildcard character "?".

DEVICE= This parameter restricts the volumes that can be assigned to this pool by supplying valid device types. If this parameter is not supplied, all device types are candidates for selection.

Specify one or a list of device types. You may also specify the devices with the following character device codes:

Table 3-17. Device Type Character Code Values

Device Type	Code
3350	B
3375	C
3330-1	D
F6425, Any capacity	E
F6427, Any capacity	F

Note: These character device codes may be used only in specifying this parameter. They are not applicable elsewhere in SAMS:Disk processing.

MOUNT= This parameter restricts the volumes that can be assigned to this pool by supplying valid mount status types. If this parameter is not supplied, all mount status types are candidates for selection.

Supply PUBLIC, PRIVATE, STORAGE in any combination for mount status selection.

VOLSER= Supply a 6-character volume serial or pattern, or combinations of both, to specify a subpool of volumes within this pool definition. Pattern characters are the wildcard "?" and the pad to the right "/". If less than six characters are specified, the pad is assumed.

THRESHOLD= You may specify one THRESHOLD= subparameter on a VOLSER= parameter, provided it follows the volume serial name or pattern, to set a different threshold for volumes in the subpool.

Implementation Considerations

Although the EXCLUDE= keyword can specify data sets to be excluded from processing, you may also wish to exempt certain volumes from processing by the Threshold Manager. Assign these volumes to a pool, then specify ARCHIVE=NO and MIGRATE=NO. This in effect prevents volumes from being selected by a subsequent pool definition.

When a pool is defined with ARCHIVE=NO and MIGRATE=YES, the DSCL Move commands will be generated with the objective of distributing the data across

the volumes in the same proportion as the relative thresholds. This is the default setting.

When a pool is defined with ARCHIVE=YES and MIGRATE=NO, data sets are selected as archive candidates based only on the threshold limit defined for the volume on which they reside. If MULTIVOL=YES, then the decision to archive is based on the threshold limit of each containing volume. If archive is required to bring any volume to its threshold, the multivolume data set will be selected, regardless of whether or not archive is required for the other containing volumes.

When a pool is defined with ARCHIVE=YES and MIGRATE=YES, the Threshold Manager processes volumes in the pool in two phases. During the first phase, candidates for archiving are selected. The threshold limit used is the average threshold of the pool (the sum of the individual volume thresholds, divided by the number of volumes in the pool). Archive candidates are selected from any of the volumes to achieve the pool's average threshold limit. If the volume pool is already at its average threshold limit, no archive commands are generated.

The second phase of processing begins once sufficient archive commands have been generated to bring the pool down to or below its threshold. Using the individual volume threshold values, data sets are selected for migrating, such that each volume will end up at or below its individual threshold.

Threshold Manager User Exit

The Threshold Manager program calls a user exit after a data set or VSAM cluster is determined to be a candidate for archive or move processing. A logical 32-bit preference weight has already been assigned as an archive preference weight. This number is the logical inverse of the right justified 3-byte last reference date. Candidates will be placed on a list in ascending order of this preference weight.

At this point the user exit can make the following decisions.

1. Reject the candidate for archiving.
2. Reject the candidate for move.
3. Reject the candidate for both archiving and moving.
4. Change the archive weight, thereby changing the placement of the candidate on the archive list. (Archive candidates are used in the order in which they appear on the candidate list until the desired threshold is attained or the list is exhausted.)

A user exit may be implemented for all processing by specifying the unique member name in the USEREXIT= parameter of the GLOBAL= keyword. At the pool level, this value may be overridden by specifying the unique member name in the USEREXIT= specify and call any user exit load module by supplying its parameter of the POOLDEF= keyword.

Any number of pools may use the same user exit module, but different pools may use different exits. If no user exit module is specified for a pool, it will use the equivalent of an KDJBR14.

The exit module will be made available to the precessor by a standard load. This means it must be located in the library specified by the JOBLIB or STEPLIB DD statement.

A skeleton user exit module is provided by the name THRESEXT and is included in the installation library. It can be compiled by any assembler compiler and needs no special macros.

This skeleton contains the layout for a candidate record and provides addressability. Your decision can be based on attributes of the data set name or cluster name that is provided in the record. After you have made your decision, you need only to branch to the label in the exit program that affects your decision.

Chapter 4. Sysparms

General Information

Each installation using SAMS:Disk to manage data storage has unique needs, requiring a variety of techniques for implementing SAMS:Disk to provide compatibility with each environment. To meet these various needs, SAMS:Disk was developed to use system parameters (sysparms) to control various aspects of each function. The default value of each sysparm may be altered by users to meet the operating requirements of each installation. Most system parameters can also be specified temporarily, through overriding dd statements, to accommodate any special circumstances of a particular job run.

Each function in SAMS:Disk has logic to access a sysparm when it is necessary to determine the value in effect. Once the value has been retrieved, program logic will invoke the code to perform the function according to the sysparm value specified.

SAMS:Disk assigns a default value to every system parameter and will execute according to this default value unless a different value is assigned through an entry in the parmlib data set member SYSPARMS or through an overriding dd statement.

The SAMS:Disk default value of a sysparm is assigned, based on the “typical” installation needs. In many cases, the default value will be the best value for your installation and no adjustments are required. Only if the default value does not meet your installation’s needs will you need to make any changes.

To change a sysparm to a value other than its default value, first create a member in the parmlib data set called SYSPARMS. This member is not provided with the system. However, a member called SAMPARMS is provided as a sample, and can be copied in to create your initial SYSPARMS member. System parameters are then specified in the SYSPARMS member, one parameter per line. If more than one entry is entered on a line, all but the first will be treated as comments.

Each system parameter is composed of a parameter name and a parameter value. The name is the first eight characters and the value is appended immediately after the name. All parameters have unique names; the length and type of data required in the value differ. The default value and the acceptable user-specified values for each sysparm are documented later in this section under each sysparm name.

To add, change or delete entries in the SYSPARMS member, use any online editor or other conventional means for updating a member of a partitioned data set.

System Parameter Names Reserved for Users

Occasionally you may need to create your own system parameters, perhaps in conjunction with a user exit or for user-supplied applications using SAMS:Disk modules. (See the *Modifying SAMS:Disk* section beginning on page 353 in this manual.) In order to ensure that sysparms you create will not conflict with SAMS:Disk-supplied sysparms in a future release, specify sysparms you create with a name beginning with a dollar sign (\$). By definition, all sysparm names beginning with a \$ are created and specified by the user or installation.

Overriding Sysparms Instream

To provide maximum flexibility, SAMS:Disk also permits the user to supply an input stream of overriding system parameters through the //SYSPARMS dd statement. Sysparms thus supplied will either supplement those specified in your SYSPARMS member of parmlib or override them if already present.

The dd statement must point to a sequential file with DCB attributes of LRECL=80 and RECFM=F or FB. It may be a dd * type input stream. Sysparms start in column 1 of the record image, as shown in the example below:

```
//SYSPARMS DD *
* COMMENTS HAVE ASTERISK IN COLUMN ONE.
ADPCTPDS025
FLYCOUNT0
PRIEXMODMYEXIT          ADD A USER EXIT
'DSDISPEX '             BLANK OUT USER EXIT
/*
```

Figure 4-1. JCL to Override SYSPARMS

The capability to override sysparms through an input stream is enabled by specifying sysparm SYSPARMO with a value of Y or S in the SYSPARMS member of parmlib. (See *SYSPARMO* description on page 188 for details.) The default value of N disables this feature.

Note that the sysparm values specified in the input stream are temporary overrides. They do not update the contents of the SYSPARMS member of the parmlib data set. To protect against unauthorized use of restricted functions or operations, certain sysparms may not be overridden by this technique.

A cross-reference matrix of all sysparms and a more detailed discussion of each, including the default value, is provided in the remainder of this section.

System Parameters

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
ADPCTOTH				•	•		•				•							•				
ADPCTPDS				•	•		•				•							•				
ADTRACKS				•	•		•				•							•				
ALLOCPRI	•	•	•													•	•	•				
ALLOEXIT				•	•		•				•											
ALLOVBLK				•	•		•				•							•				
ALOCATEX				•	•		•				•						•	•				
ALOPOSEX				•	•						•						•	•				
ALTTRKAS				•	•	•					•							•				
ARBACKUP			•																•			
ARCATOPT		•	•																			
ARCBLSI		•	•																			
ARCCNAME		•	•			•																
ARCCTYPE		•	•			•																
ARCDSKBZ		•	•																			
ARCDSNEX		•	•			•																
ARCDSORG		•	•																			
ARCEMPTY		•	•																			
ARCEODEX		•	•			•																
ARCFDEFR		•	•																			
ARCHIVEX		•	•																			
ARCHPACK		•	•	•	•		•		•	•	•											

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
ARCHSORT		•	•			•																
ARCMODEL																						
ARCMULTI		•	•																			
ARCRETPD		•	•																	•		
ARCSEPC0		•	•																			
ARCTNAME		•	•			•			•	•												
ARCVOLID		•	•																			
ARC0NAME		•	•			•																
ARC0TYPE		•	•			•																
ARDDNEXC				•							•											
ARDIAGNM				•							•											
ARDSNPOL				•							•											
ARESCRDT				•							•											
ARESPREX				•							•											
ARESUNIC				•							•											
ARESUNIT				•							•											
ARESUNIn				•							•											
ARESTENQ				•							•											
ARJOBEXC				•							•											
ARPOOLEX				•							•											
ARSECURE				•							•								•			
ARTAPEOK				•							•											
ARTASKNM				•							•											
ARVLPOL				•							•											

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
CARTFEET		•	•							•						•						
CATBKDEL					•		•															
CATBKNAM					•		•															
CATBKSPC					•		•															
CATBKUNT					•		•															
CATLOCWA																						•
CATPURGE		•	•																			
CNVATCYL				•	•		•				•											
DARCHIEX		•	•																	•		•
DARCKEEP		•	•																	•		
DASDPMAX				•	•						•											
DASDVMAX				•	•		•				•											
DATECUTO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DBDSNCFD																					•	
DBDSNINX																					•	
DCAVGCMP		•	•																			
DCCOMPTC		•	•			•																
DCDATAACP		•	•		•	•	•															
DCDSNDEX		•	•																			
DCEXCTBL		•	•			•																
DCINCTBL		•	•			•																
DCRTSTAT		•	•																			
DEFARCEX																						
DELETEOK		•	•																			

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
DERASEEX		•	•	•							•									•		
DFLTPOOL				•	•		•				•											
DIAGALOC				•	•												•	•				
DIAGAUTH	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DIAGEXCP		•	•	•	•		•				•							•				
DIAGVSRL															•							
DINXUFEX		•	•	•	•	•	•				•											
DISKPROT		•	•			•				•												
DLVOLMSG	•																					
DRESARCH				•							•											
DRESKEEP				•							•									•		
DRESTOEX				•							•									•		•
DSBALTRT																					•	
DSBCATSZ																					•	
DSBSELVS																					•	
DSBTRKSZ																					•	
DSBUNITS																					•	
DSBUSREX																					•	
DSBVOLCT																					•	
DSCBCVEX				•	•		•				•						•	•				
DSCBJBNM	•	•	•	•							•											
DSCBLMOD	•																					
DSCBLUSD	•																	•				
DSCBOPCD	•																					

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
DSCBSVMD	•			•	•						•						•	•				•
DSCBUSRD																				•		
DSCLMSGs	•	•	•			•									•			•				
DSCLOWRC		•	•			•									•			•				
DSCLSCEX	•	•	•	•		•					•				•							
DSDISPEX		•	•																			
DSNDELEX									•													
DSNDLPEX									•													
DSNDUPVL				•	•				•		•			•								
DSNSELRE	•	•	•			•									•			•			•	
DSUTILCF	•																					
DSUTILSQ	•																					
DYNCUNIT		•	•	•	•		•			•	•											
DYNEXPDT		•	•			•				•												
DYNTUNIT		•	•	•	•		•			•	•											
DYNUKEEP		•	•	•	•	•	•			•	•											
DYNVREQD		•	•							•												
DYNnUNIT		•	•	•	•		•			•	•											
ENQADDTL		•	•	•	•	•	•				•				•	•	•	•				
ENQDSSIM																	•					
ENQGDGBY		•	•	•	•	•	•				•								•			
ENQMSGES		•	•	•	•						•				•	•	•	•				
ENQSTPOK	•	•	•	•	•	•	•				•				•	•	•	•		•		•
ENQVOLWT		•	•	•	•		•			•	•	•										

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
ENQWAITD	•	•	•	•	•	•	•				•				•	•	•	•		•		•
EXPARSOP		•	•																			
EXPASCOP		•	•																			
EXPDTUSE		•	•																			
FASTCOMP																	•					
FILESCNT														•								
FILESPEC		•	•	•	•	•			•	•	•	•	•	•		•			•	•	•	•
FILEUNIT		•	•	•	•	•			•	•	•	•	•	•		•			•	•	•	•
FILOGNAM		•	•	•	•	•			•	•	•	•	•	•		•			•	•	•	•
FLYCOUNT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
FMSSAFEX				•	•						•											
FMSSPREX				•	•						•											
FORCPool		•	•						•	•			•									
GDGCATEX				•	•						•						•	•				
GDGDSNCT																•						
HONORTTR				•	•		•				•							•				
ICHEKAEX		•	•	•	•		•				•						•	•	•			
ICHEKPEX		•	•	•	•		•				•						•	•	•			
IDRCUSED		•	•			•				•												
IGNLSTAR		•	•															•				
IOERRLIM						•																
IOMAXREC		•	•	•	•	•	•				•											
IOTRACKS		•	•	•	•		•				•							•				
ISAMRECO		•	•															•				

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
IXCTLGEX																						
IXMCLEAN									•					•								
IXMRESEX									•													
IXMVOLON									•	•												•
IXUSRNOT									•													
LISTDMEX																				•		
LISTDVEX								•												•		
LISTREEX																				•		
LISTSORT				•																		
LISTVTOC					•																	
MERCnNAM										•												
MERCnTYP										•												
MERDSNEX										•												
MERFASTS										•												
MERPnNAM										•												
MERPnTYP										•												
MERREPLY										•												
MERVOLX										•												
MIGBYPAS																•						
MIGCATLG																•						
MIGCREDT																•						
MIGCTGDG																•						
MIGDISEX																•						
MIGDSMAX																•						

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
MIGDSORG																•						
MIGEXPDT																•						
MIGFXPDT																•						
MIGGDGEX																•						
MIGMODEL																•						
MIGNOEXP																•						
MIGNXPDT																•						
MIGPSCOP																•						
MIGREXIT																•						
MIGRSORT																•						
MIGYRCHG																•						
NOBLKMSG																	•					
OFFLDASD				•							•											
PASSNEWN				•							•						•					
PASSWORD	•	•	•	•	•	•	•				•				•	•	•	•		•		
PDSANMLY		•	•	•	•		•				•											
PDSENQSH																	•					
PDSEXSIM																	•					
PDSMAXPR																	•					
PDSMAXSE																	•					
PDSMINDB																	•					
PDSMINPR																	•					
PDSMINSE																	•					
PDSRRMAX																		•				

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
PDSSETBS																	•					
POINTABS				•	•		•			•	•											
PREALLOC				•	•		•				•											
PRIALLOC				•	•						•						•	•				
PRTSORTF	•																					•
PSUDEVTP		•	•																			
RAADDAEX																	•	•	•			
RAADDPEX																	•	•	•			
RACFALLO				•	•		•				•							•	•			
RACFALWZ	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•	•		•
RACFBITS				•	•						•						•	•	•			•
RACFBKUP																			•	•		
RACFDAEX																			•	•		•
RACFDPEX																			•	•		•
RACFDVL2																	•					
RACFDVOL		•	•	•	•	•	•	•	•	•	•								•	•		•
RACFEAEX		•	•	•	•	•	•	•	•	•	•								•			•
RACFENTY		•	•	•	•	•	•		•	•	•								•	•		•
RACFEPEX		•	•	•	•	•	•	•	•	•	•								•			
RACFMDSN				•	•		•		•		•								•			
RACFMODL				•	•		•		•		•								•			
RACFMVOL				•	•		•		•		•								•			
RACFNEWN		•	•	•		•					•		•					•	•	•		•
RACFPDSW																	•		•			

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDN	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
RACFPRED				•	•		•				•							•	•			
RACFPROC	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•		•	
RACFSEQM																•			•			
RACFSEQT																•			•			
RACFSMEX																•			•			
RACFSUPP	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•		•	
RACFUSID	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•		•	
RACFVCAV																		•	•			
RACGDGVL	•	•	•	•	•	•					•				•	•	•	•	•	•		•
RACHKAEX	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•		•	
RACHKPEX	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•		•	
RADEFAEX	•	•	•	•	•	•	•	•		•	•			•			•	•	•	•		
RADEFPEX	•	•	•	•	•	•	•	•		•	•			•			•	•	•	•		
RADELAEX									•	•									•			•
RADELPEX									•	•									•			•
RADELSVC																			•			•
REBLIMIT																		•				
REBLKBIT				•	•						•							•				
REBYPASS																						
RECATPSU		•	•																			
RECATVOL		•	•	•	•		•		•		•											
RECCHCAT					•		•															
RECPRIEX					•																	
REDEVDEF		•	•	•	•		•				•											

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
RESAFTEX				•	•		•				•											
RESCHCAT				•							•											
RESCOPYT				•	•		•				•											
RESCRCAT				•	•						•											
RESIXDLB				•	•		•		•	•	•											
RESIXRPD				•	•		•		•	•	•											
RESPRIEX				•	•		•				•											
RESRETPD				•	•		•				•											
RESTCTLG				•	•		•				•											
RETAFTTEX																						
RETDSNCD																						
RETDSNCL																						
RETDSNDF																						
RETDSNLU																						
RETEXCLD				•	•		•				•											
RETEXCMD				•	•		•				•											
RETGDGEX																						
RETGDGNM																						
RETMULTI																						
RETNODSP																						
RETNOMIG																•						
RETPREFT																						
RETPRIEX																						
RETRETPD						•																

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
RLSEAFEX															•							
RLSEDIAG															•							
RLSEMAXS															•							
RLSEMINS															•							
RLSESECR															•							
RLSESYS1															•							
RLSEUNMV															•							
RPTCLSEL																						
RPTDSBEX	•																					
RPTDSNEX	•																					
RPTDSUEX	•																					
RPTLINES	•																•					
RPTMVDEX	•																					
RPTNMTSZ	•																					
RSCHGBIT				•	•						•											
RSUPPRES									•													
SCRASYS1		•	•	•	•		•				•					•	•	•		•		
SCRTCHEX		•	•	•	•		•				•					•	•	•		•		
SECURAEX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
SECURCMD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	
SECURDRS				•							•											
SECURFIL		•	•	•	•	•	•	•	•	•	•	•	•	•		•			•	•		•
SECURLIM	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•			•	•		•
SECURLOG	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDN	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
SECURPEX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
SECURRPT	•							•											•	•		
SECURVOL	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•		
SECUSUPP	•	•	•	•	•	•	•				•				•	•	•	•		•		•
SELECTRE	•	•	•			•									•			•			•	
SHRNKFDT		•	•	•	•	•	•				•											
SHRNKSYS		•	•	•	•						•											
SMFCDATD	•																					
SMFCTIMD	•																					
SMFDFLAG	•																					
SMFDSNMD	•																					
SMFJDATD	•																					
SMFJTIMD	•																					
SMFTYP14	•																					
SMFTYP15	•																					
SMFVLSRD	•																					
SORTPADD	•																					
SPACECEX				•	•		•				•							•				
SPACEPRI		•	•							•												
SPACESEC		•	•							•												
SPFAUNIT																						•
SPFDSNAM	•																					•
SPFENQQN	•																					•
SPFENQTY	•																					•

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
SPFENQWT		•	•	•	•		•				•											
SPFHDGEX	•																					•
SPFHILVN																						•
SPFORDEX																						•
SPFORSEX																						•
SPFORSPC																						•
SPFORSPE																						•
SPFRPTDT																						•
SPFSYSIN																						•
SPFUSRID																						•
SVCNODMS		•	•	•	•		•				•	•			•	•	•	•		•	•	
SYSOUTEX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SYSPARMO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SYSPARMS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
TAOPPOEX		•	•			•												•				
TAOPPREX		•	•			•												•				
TAPCTLEX									•	•		•	•									
TAPEDLEX									•	•												
TAPEFEET		•	•							•												
TAPEFSCR		•	•			•				•												
TAPEPOOL		•	•			•			•	•										•		
TAPEPROT		•	•			•				•												
TAPEPRTR		•	•							•												
TAPEVVER		•	•			•				•												

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
TAPMFEET																•						
TERMCLER																				•		
TMSCTLEX									•	•			•									
TPMGMTX									•	•		•	•									
TPOOLCHG																						•
TPOOLSEQ		•	•						•	•												•
TRKOVCHG																		•				
TSODACAT																				•		•
TSODARES																				•		•
TSODASCR																				•		•
TSOTULMT																				•		
TSOTUNIT																				•		
TSOUSRID																				•		
TSOVOLOF																				•		
TTRULMOV																		•				
UCBALLOC	•	•	•	•	•		•			•	•	•			•	•	•	•		•	•	•
UCC1BYPS									•	•		•	•									
UCC1EXPR									•	•		•	•									
UCC1EXTN									•	•		•	•									
UCC1SCRT									•	•		•	•									
UNCATARC									•	•												
UNCATDEL		•	•			•		•	•	•												
UNCATFDR									•	•												
UNCATPSU									•	•												

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTD/V	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
UNLOADEX														•								
UNMOVEAR		•	•	•	•		•				•											
UNMOVEVC																		•				
USERSEQ		•	•	•	•	•	•				•				•	•	•	•				
USERVENQ						•																
USRCLEN																					•	
USRCODEX																					•	
VCBYPASS																		•				
VCCHGBIT																		•				
VCCOPYEX																		•				
VCDMIGEX																		•				
VCUDSORG																		•				
VOLSELRE	•	•	•			•									•			•			•	
VSACCESS		•	•																			
VSACESEX		•	•																			
VSAIXDEF				•	•		•				•											
VSALLOEX				•	•		•				•											
VSALOSHR		•	•	•	•		•				•											
VSAMSUPP		•	•																			
VSAPPEND				•	•		•				•											
VSARCAIX		•	•																			
VSARCFMT		•	•	•	•		•				•											
VSATTREX				•	•		•				•											
VSATTRIB				•	•		•				•											

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
VSBIXCAT				•	•		•				•											
VSBIXPSW				•	•		•				•											
VSBLDINX				•	•		•				•											
VSCATMAX																						
VSCATPTN																						
VSCIACAT	•																					
VSCIADET	•																					
VSCOMPNM	•																					
VSCRIMAX																						
VSDATEFT				•	•		•				•											
VSDATSUF				•	•		•				•							•				
VSDEFCAT				•	•		•				•											
VSDSPACE	•	•	•															•				
VSIDCUTP				•	•		•				•											
VSIDCUTV				•	•		•				•											
VSINXSUF				•	•		•				•											
VSMFTEX	•																					
VSMPRIEX	•																					
VSMPURGE		•	•	•	•		•				•											
VSMMSGEMP		•	•																			
VSMMSGTOT		•	•																			
VSMMSGWRN		•	•																			
VSMSPWEX	•	•	•	•	•		•				•											
VSMSTATS																						

Sysparms	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDN	IXMAINT	MERGE	XCOPY	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD
VSNEWNEX				•	•		•				•											
VSNOERAS				•	•		•				•											
VSNOSCRO				•	•		•				•											
VSONLINE																						
VSPREDEF				•	•		•				•											
VSRESCLD				•	•		•				•											
VSSEQCNT				•	•		•				•											
VSSEQERR		•	•																			
VSSETUSE				•	•		•				•											
VSSPEEDO				•	•		•				•											
VSSPUSED				•	•		•				•											
VSSTUMPL	•	•	•	•	•		•				•											
VSSYSVER		•	•																			
VSUSEBLK																						
VSUSEMOD																						
VSVERIFY		•	•												•							
VTOCREAD	•						•								•	•				•	•	
WTORESRL																	•					

Sysparm Definitions

To place in effect a value other than the default value for any of the following system parameters, create a one-line entry in member SYSPARMS of the parmlib data set, enter the name of the sysparm and append the appropriate value.

ADPCTOTHnnn

Specify a numeric value from 000 to 100 to represent the percent increase to be added to the SAMS:Disk-computed space value when allocating a non-PDS data set to a different device type than it was archived from. This percentage value is added after the SAMS:Disk-computed value, allowing for device type changes and any other command parameters that apply to space conversion. The default value is 000.

ADPCTPDSnnn

Specify a numeric value from 000 to 100 to represent the percent increase to be added to the SAMS:Disk-computed space value when allocating a PDS data set to a different device type than it was archived from. This percentage value is added after the SAMS:Disk-computed value, allowing for device type changes and any other command parameters that apply to space conversion. The default value is 000.

ADTRACKSnnnn

Specify a numeric value from 0000 to 9999 to represent the number of tracks to be added to the SAMS:Disk-computed space value when allocating both VSAM and non-VSAM data sets to a different device type than they were archived from. For a VSAM KSDS, the tracks will be added to both the data and index components. The default value is 0000.

ALLOCPRly

When a data set is archived (or backed up), a “space allocated” value is saved in the DSNINDEX record and can subsequently be displayed. This sysparm indicates whether the size of the first extent (primary allocation value) or the sum of all extents (total allocation) is to be saved. The default value for this sysparm is Y (save the primary allocation). A value of N causes the total allocation to be saved.

Several of the SAMS:Disk reports have both the PRIM field (first extent value) and an ALLOC field (total allocation value). Setting this sysparm to N will cause the PRIM field to be equivalent to the ALLOC field, which is probably not very useful. Therefore it is recommended that a value of N be used only for backup or archive processing.

ALLOEXITmmmmmmmmmm

Specifying this sysparm causes SAMS:Disk to call a user exit. The module named by “mmmmmmmmmm” is given control at data set allocation time during restore or recover processing. The format-1 DSCB of the data set to be allocated is passed to the module to allow the CREDIT, EXPDT, last used date, last MOD date, or open count fields to be reset. Turn to the user exit description for *ALLOEXIT* on page [221](#) for more information.

ALLOVBLKn

Specify this sysparm with a value of Y to allow data sets with no track overflow and a blocksize larger than the track capacity to be allocated on the volume. The default is N, which causes SAMS:Disk to not allocate the data sets.

ALLOCATEXmmmmmmmmmm

Specify this sysparm with the module name to be invoked for the *ALLOCATEX* user exit. Turn to the user exit description for *ALLOCATEX* on page [221](#) for more information.

ALOPOSEXmmmmmmmmmm

Specify this sysparm with the module name to be invoked for the *ALOPOSEX* user exit. Turn to the user exit description for *ALOPOSEX* on page [222](#) for more information.

ALTTRKASy

This sysparm controls the assignment of alternate tracks if an error is encountered during a write to disk operation. The default value for this sysparm is Y, which allows alternate tracks to be assigned. Specify a value of N to prevent assignment of alternate tracks.

ARBACKUPy

This sysparm allows you to determine which DSNINDEX records may be candidates for autorestore. This sysparm may be useful in 2 situations:

1. If you use the MFILES DD statement and considerations regarding order of concatenation is important
2. Whether backup and archive copies are kept in separate files data sets

There are 3 possible values for this sysparm:

Table 4-1. Possible Values for ARBACKUP

Value	Description
Y	This default value allows the first DSNINDEX record found to be used. Users who ensure the latest copy of a data set will be found first in the MFILES concatenation will normally use this setting. Users who use the DSCL DELETE DISP=RECAT command may also see benefit in this setting.
N	Will only allow archive copies to be considered, and will ignore or skip any backup DSNINDEX records as MFILES are searched.
1	Will scan for the first DSNINDEX record found, but will insist that that first record ALSO be an archive copy. If it is not, the restore will fail. This ensures that only the latest archive copy will be used when MFILES are expected to have archive copies found before any backups. This might be the case in shops emulating DFHSM or similar processing.

N

ote: This sysparm will add a TIMEDATE= parameter to the generated autorestore command, and this TIMEDATE will be used for Immediate or Deferred requests. In previous releases of SAMS:Disk, DISPOSE would use the first available DSNINDEX at the time the DISPOSE job executed.

ARCATOPTu

The default value of this sysparm is U, which causes catalog action in batch explicit archive (not retention control or VSAM implicit processing) to uncatalog the data set. Specify a value of R to recatalog to the pseudo-volume serial ARCIVE. Specify a value of N to use no-catalog action. Any of these options may be overridden by the RECATALOG, UNCATALOG or NOCATALOG parameters on the ARCHIVE command.

ARCBLKSIn

The default value of this sysparm is N, which causes SAMS:Disk to bypass processing for data sets with zero blocksize. Specify a value of Y to allow data sets with zero blocksize to be archived. They will be processed as physical sequential data

sets. For data sets with no extents (model DSCBs), sysparm ARCMODEL will take precedence, even when the blocksize is zero.

ARCCNAMEnnn

If MSP dynamic allocation is used to allocate the //ARCHIVEC dd statement (for the duplicate copy being written to the archives), this sysparm provides the data set name to be used. The nnn value is the 1- to 22-character base name, the complete name or the tape GDG index, depending on installation JCL and other sysparm options. The default value is DMS.ARCHCOPY. See “*Archive/Backup Considerations*” on page 57 for more information on the generation of this data set name.

ARCCTYPEtape

If the //ARCHIVEC dd statement is not provided in the JCL, this sysparm indicates the type of device to be dynamically allocated. It must specify one of the following: TAPE, F6470, DISK, DYN1, DYN2, DYN3 or NULL. TAPE is the default.

ARCDSKBZbbbbbb

SAMS:Disk assigns an optimal blocksize to an archive data set being written to disk, based upon the disk device type. To override the computed value, specify the desired blocksize for the bbbbbbb value. The default of 00000 indicates that the computed value is to be used.

ARCDSNEXmmmmmmmm

This sysparm indicates the module to be given control after a SAMS:Disk archive data set name is generated. Turn to the user exit description for ARCDSNEX on page 223 for more information.

ARCDSORGn

Specify this sysparm to allow data sets that have no DSORG specified in the format-1 DSCB to be eligible for processing by either archive or backup functions. The default is N, which causes SAMS:Disk to bypass data sets that have no valid DSORG specification.

ARCEPTYy

Specify this sysparm with a value of N to suppress the archival of empty data sets. The default is Y, which causes SAMS:Disk to archive empty data sets if they are selected for processing.

ARCEODEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ARCEODEX user exit. Turn to the user exit description for ARCEODEX on page 224 for more information.

ARCFDEFRn

Specify this sysparm with a value of Y to queue a deferred archive request for a data set that would otherwise be bypassed when an enqueue cannot be obtained. The

default value is N, which causes SAMS:Disk to bypass the data set when an enqueue cannot be obtained. This sysparm is not intended to requeue requests during dispose processing.

ARCHIVEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ARCHIVEX user exit. Turn to the user exit description for *ARCHIVEX* on page [225](#) for more information.

ARCHPACKy

Specify this sysparm to control archive data set packing. The default value is Y, which causes each archive data set to contain multiple data sets. A value of N causes SAMS:Disk to create one archive data set for each data set processed. This sysparm controls data set packing for both the merge and archive functions.

Warning: For archival to tape or cartridge, do not specify a value of N, or every archive tape or cartridge will contain only one data set.

For further discussion of this sysparm, see “*Using Archive Functions to Manage Active Data*” on page [67](#) in this manual.

ARCHSORTudv

Specify this sysparm to change the selection of archive reports produced. The default value of UDV will cause archive reports to be sorted by different fields as described below:

Table 4-1. Possible Values Available for ARCHSORT

Value	Description
U	print unsorted archive report
D	print archive report in dsn/volume sequence
V	print archive report in volume/dsn sequence
N	Specify N for any report not desired

ARCMODELn

Model DSCBs may be archived (and restored) during archive processing if this sysparm is specified with a value of Y. The default value is N.

ARCMULTIn

Specify a value of Y for this sysparm if multivolume PS and DA data sets are to be processed by explicit archive requests processed by the ARCHIVE proc. The default value is N, not to process multivolume data sets.

ARCRETPDnnnnn

This sysparm controls the number of days that an explicitly archived data set is to be retained in the archives. The default value is 00030. You may specify any 5-digit value. If the user has specified the RETPD parameter on the ARCHIVE command, its value will override the one specified here.

ARCSEPC0y

When both the primary and the duplicate copy archive data sets are being written to disk, the default value for this sysparm is Y, which instructs SAMS:Disk to always separate the copy from the primary; that is, never put the backup copy on the same disk volume as the primary. Specifying N (no separation) permits allocation to the same volume.

ARCTNAMEd

By specifying one of the 4 values for this parameter, you may tell SAMS:Disk how to generate archive tape data set names, and whether or not they are to be cataloged.

Table 4-2. Possible Values Available for ARCTNAME

Value	Meaning
D	generates a unique tape data set name but not catalog it (default value)
C	generates a unique tape data set name and catalog it
G	will use GDG support to generate a unique tape data set name and catalog it
U	uses data set name specified in JCL, or the exact value of sysparm ARCxNAME

Options D (default value) or C are preferable to G or U and should be used wherever possible. These options generate unique data set names for each tape data set created. They provide significantly enhanced flexibility if you are using tape management programs such as CA1, and eliminate data set name contention during allocation. Option C also catalogs the tape data set names. For more information about the benefits of using these options, see *"Cataloging the Tape Archives"* on page 41 and *"Naming the Archives"* on page 61 in this manual.

Setting this sysparm to a value of G provides the benefits of generating unique tape data set names, requires building of GDG indexes, and is limited to 255 generations per index. It is maintained strictly to provide compatibility with prior releases.

Setting this sysparm to a value of U forces SAMS:Disk to create output archvols based on either:

1. The value specified for DSN= in JCL

2. The value specified for ARCxNAME

The value of U is not allowed in Merge due to allocation contention if the same dsname were to be mounted as both input and output. Its also not permitted in Archive/Backup if the same dsname is specified for both ARCxNAMEs.

If you have specified either a value of C or G, you should also specify sysparm UNCATARC with a value of Y to tell SAMS:Disk to uncatalog the archive data set when it expires the volume.

If your security system protects tape volumes and you wish to specify options C or G for this sysparm, you must also ensure that all users running backup or archive jobs have access to tape volume DMSTMP for the following reason.

To serialize tape names, SAMS:Disk will temporarily catalog the data set name to the fictitious volume DMSTMP. SAMS:Disk will then open the tape, determine the actual tape volume that was mounted, and then add it to the catalog entry, replacing the temporary value of DMSTMP. For this catalog action to take place correctly, the user running SAMS:Disk must have authority to tape volume DMSTMP.

ARCVOLID@#\$+-%&[

When archival to disk is being used, the key to the ARCHVOLS record is generated by SAMS:Disk. Each key is built by taking the first character of this sysparm and appending 99999 to it. Each successive key is decremented by 1 to make it unique. When 00000 is reached, the second character from this sysparm is used and the process started over.

The default value for this parameter is @#\$+-%&[. You may change the order, or specify other special characters. However, you must not specify any characters that are used in volume serial numbers at your data center, or any of the SAMS:Disk pattern characters (*,?,/,!). Valid characters are:

[+] & \$; ^ - | % _ : # @ ~ \

ARC0NAMEnnn

If MSP dynamic allocation is used to allocate the //ARCHIVE0 dd statement (for the primary copy being written to the archives), this sysparm provides the data set name to be used. The nnn value is either the 1- to 22-character base name, the complete name or the tape GDG index, depending on installation JCL and other sysparm options. The default value is DMS.ARCHPRIM. See “*Archive/Backup Considerations*” on page 57 for more information on the generation of this data set name.

ARC0TYPEtape

If the //ARCHIVE0 dd statement is not provided in the JCL, this sysparm indicates the type of device to be dynamically allocated. It must specify one of the following: TAPE, F6470, DISK, DYN1, DYN2 or DYN3. TAPE is the default.

ARDDNEXC_ddname

This sysparm is intended to assist with the IVP process of the test auto-restore function (TSTAR). For the value of this sysparm, specify the DD statement name you want to invoke TSTAR. Jobs displaying this DD statement will have explicit access to the TSTAR started task. For related information, please review the sysparm description for *ARJOBEXC* on page 131 in this manual.

The default value for this sysparm is _DDNAME.

ARDIAGNM_jobname

This sysparm is intended for debugging purposes of the auto-restore function and is inactivated by default, which provides no diagnostic information. For the value of this sysparm, specify the jobname in which you want to invoke these diagnostics.

The default value for this sysparm is _JOBNAME.

ARDSNPOLxxxxxxx

Use this sysparm to identify to SAMS:Disk the name of the member in your parmlib that contains the data set name pool for auto-restores. If you choose to use pool support for Auto-Restores, we recommend that you use the name ARDSNPOL for your member. To specify this, code the sysparm as ARDSNPOLARDSNPOL. See the topic *"The Auto-Restore Function"* on page 42 in the *Installation Guide* for more information regarding pool support.

ARESCRDTn

This parameter, which is valid for Auto-Restore only, allows selection of the date to be used for the data set catalog and format-1 creation date. This parameter provides the same capability as the CREATE parameter does for Restore. The default value of N sets the catalog and format-1 creation dates to the original date. Specifying a value of Y sets the catalog and format-1 creation dates to the current date.

ARESPREXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ARESPREX user exit. Turn to the user exit description for *ARESPREX* on page 226 for more information.

ARESUNICxxxxxxx

Specify for "xxxxxxx" the 1- to 8-character unit name for the class of F6470 devices to be used for dynamic allocation of SAMS:Disk archive cartridges in the auto-restore function. The default unit name is "CART". A specific unit address may also be supplied to force all auto-restores from cartridges to be performed on a single F6470 device.

ARESUNITxxxxxxx

Specify for "xxxxxxx" the 1- to 8-character unit name for the class of tape devices to be used for dynamic allocation of SAMS:Disk archive tapes in the auto-restore func-

tion. The default unit name is TAPE. A specific unit address may also be supplied to force all auto-restores from tapes to be performed on a single tape device.

ARESUNInxxxxxxxx

Specify a 1- to 8-character unit name associated with special device type DYN1 (2 or 3). This unit name is used for dynamic allocation of archive volumes during auto-restore whenever the DYN1 (2 or 3) flag is on in the ARCHVOLS record. (This flag will have been set by prior archive or merge runs that created the archive volume specifying DYN1 (2 or 3) for sysparm ARC0TYPE, ARCCTYPE, MERPhTYP, OR MERChTYP.) A specific unit address or an esoteric name that maps only to the special devices is recommended as a value for these sysparms. They default to unit names DYN1, DYN2, and DYN3, which are probably invalid for your installation, and therefore need to be changed to the unit names as generated for your system.

These sysparms are identical in purpose to DYN1UNIT, DYN2UNIT and DYN3UNIT, but are used exclusively by the auto-restore function.

ARESTENQy

Specify 'Y' or 'N' to indicate if DMSAR is to perform data set enqueues or not, respectively. The default value is 'Y'. Such enqueues have QNAME='SYSDSN' and RNAME=*dsname*.

CAUTION: The SYSDSN enqueue is MSP' way of serializing access to a data set. Disabling SYSDSN enqueues is usually not advisable. However, some installations may want to do this in some unusual circumstances.

ARJOBEXC_jobname

This sysparm is intended to assist with the IVP process of the test auto-restore function (TSTAR). For the value of this sysparm, specify the job name or task name (i.e., userid) you want to have invoke TSTAR. All submissions of jobs or tasks by this name will have explicit access to the TSTAR started task. For related information, please review the sysparm description for *ARDDNEXC* on page 129 in this manual.

The default value for this sysparm is _JOBNAME.

ARPOOLEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ARPOOLEX user exit. Turn to the user exit description for *ARPOOLEX* on page 227 for more information.

ARSECUREn

This sysparm allows you to specify the level of access a user must have to restore a data set using the SAMS:Disk auto-restore function. It can be used only if your installation has RACF installed, and has also installed and activated the corresponding SAMS:Disk security interface.

Since the SAMS:Disk auto-restore started task cannot be identified by a job statement or a LOGON procedure, most security packages have a Started Procedures Table to associate started task names with user IDs and possible group names. Most users with security packages supply an entry in this table to associate the SAMS:Disk auto-restore started task name with a user ID with unlimited access to data sets in your shop. However, these users will find that any system user can auto-restore any user data sets in the SAMS:Disk archives. Although this causes no security exposure (after an auto-restore completes, OPEN, SCRATCH, RENAME or KQCAMS processing prevents user access to data sets to which they are not authorized), this can cause confusion. This is the case if the default value, N, is in effect.

Specify this sysparm with a value of Y to have the SAMS:Disk auto-restore function convert the user ID (and group name, if it exists) of the SAMS:Disk auto-restore started task to that of the originating user. This means that to auto-restore a user data set in the SAMS:Disk archives, a user would need the same access to the data set as would be required by a batch restore run by the same user: ALLOCATE, and open for OUTPUT.

Specify this sysparm with a value of R to have the SAMS:Disk auto-restore function convert the user ID (and group name, if it exists) of the SAMS:Disk auto-restore started task to that of the originating user and check for INPUT authority. If the user is authorized, SAMS:Disk will convert back to the original user ID (and group name, if it exists). If the entry in the Started Procedures Table gives sufficient authority, the auto-restore task can then allocate the data set and open it for OUTPUT (to put the data in), even if the originating user has only INPUT authority. This means that to auto-restore a user data set in the SAMS:Disk archives, a user would need INPUT authority to the data set.

To minimize user confusion, we recommend that you set this sysparm to a value of R.

ARTAPEOKy

The default value of Y allows tape mounts to be requested for the auto-restore facility. A value of N disallows all auto-restore requests that would require a tape mount. A value of B allows tape mounts only if a batch program invoked auto-restore — it would reject any request from a TSS user. A value of D allows tape mounts for both batch and TSS users, but changes immediate restore requests from TSS to deferred requests if a tape mount is needed.

Note that this parameter has no effect on data sets that have been archived to disk. They may always be restored unless the user screening exit specifically rejects the request.

ARTASKNMdmsar

This sysparm indicates the started task name to be used by the Auto-Restore facility. The default value is DMSAR. The value of this sysparm takes effect the moment the started task is installed. If you want to use a different name for

Auto-Restore, update the value of this sysparm to reflect the new name, and issue a REFRESH.

The syntax of the REFRESH command is as follows:

```
S DMSAR ,DMSAR=REFRESH
```

ARVOLPOLxxxxxxxx

Use this sysparm to identify to SAMS:Disk the name of the member in your parmlib data set that contains the source volume pool for auto-restores. If you choose to use pool support for auto-restores, we recommend that you use the name ARVOLPOL for your member. To specify this, code this sysparm as ARVOLPOLARVOLPOL. See the topic *"The Auto-Restore Function"* on page 42 in the *Installation Guide* for more information regarding pool support.

CARTFEETnnnn

The default value for this sysparm is 0494, approximately the uncompressed size of a 200MB F6470 cartridge.

The F6470 cartridge tape is 495 feet long. Since the ends can not be "clipped", the SAMS:Disk default is to use 494 feet. If predictions indicate that more than 494 feet will be needed, a new cartridge will be requested.

CATBKDELn

When an EDF-Catalog is being recovered but the "damaged" catalog is still on disk, the damaged version will be overlaid only if its "export temporary" flag is on. If this flag is not on, the catalog must be deleted before it can be recovered. The default value of N for this sysparm causes SAMS:Disk to bypass the recovery under this condition; that is, SAMS:Disk will not delete the catalog. It must be deleted manually before rerunning the recovery.

Specify the sysparm with a value of Y to allow SAMS:Disk to delete an existing EDF-Catalog in order to continue with the recovery.

CATBKNAMExxxxxxxxxxxxxxxx

When EXPORTF=YES has been specified for recovering an EDF-Catalog, the SAMS:Disk backup copy is written to a sequential data set on disk that you can then use as direct input to a native KQCAMS IMPORT. This sysparm specifies a base name for SAMS:Disk to use to generate the data set name when allocating the sequential data set. The base name may be from 1 to 19 characters in length. SAMS:Disk appends "EXPTCAT" and the date and time to the base name to make it unique.

The default value is DMS.CATEXPT, which results in the name "DMS.CATEXPT.EXPTCAT.Dxxxxxx.Txxxxxx". As an example, if this sysparm were specified with the value "AAA.BBB.", the resulting name would be "AAA.BBB.EXPTCAT.Dxxxxxx.Txxxxxx".

CATBKSPCnnn

When EXPORTF=YES has been specified for recovering an EDF-Catalog, the SAMS:Disk backup copy is written to a sequential data set on disk that you can then use as direct input to a native KQCAMS IMPORT. This sysparm specifies the number of blocks to allocate for the sequential data set. Both the primary and secondary quantities are allocated in blocks, each block being 2048 bytes. The default value is 500.

CATBKUNTunitname

When EXPORTF=YES has been specified for recovering an EDF-Catalog, the SAMS:Disk backup copy is written to a sequential data set on disk that you can then use as direct input to a native KQCAMS IMPORT. This sysparm specifies the unit-name SAMS:Disk is to use when allocating the sequential data set. The default is SYSALLDA.

CATLOCWAnnnnnnn

This sysparm indicates the maximum number of bytes of memory to provide to Catalog Management for catalog work space when SAMS:Disk backs up EDF-Catalogs, creates a selection list of cataloged data sets, or processes a FIND command with a DSN parameter containing a SAMS:Disk pattern name.

SAMS:Disk will start with a work area of 20,000 bytes. If the initial work area is not large enough for Catalog Management, SAMS:Disk will create a larger work area based on the requirements returned by Catalog Management, then retry the request.

SAMS:Disk will not create a work area larger than the size specified by this sysparm. The default value of 0262144 allows SAMS:Disk to use up to a quarter megabyte if necessary. You may specify any other 7-digit number consistent with the memory and region constraints at your installation.

Keep in mind that SAMS:Disk also needs region for programs, I/O buffers and the like, and that most installations have more restrictive region constraints on TSS or PFD sessions than on batch jobs or started tasks.

CATPURGE_n

This sysparm indicates whether or not an unexpired catalog entry may be purged. VSAM catalogs allow an expiration date to be placed in the catalog entry for non-VSAM entries as well as VSAM. An unexpired entry causes an uncatalog request to fail, unless “purge” has been requested. Specify the sysparm with a value of Y (the default is N) to cause SAMS:Disk archive processing to purge a catalog entry even if it is found to be unexpired.

Fujitsu message JBB331I 084-000 will still appear in the JES log to indicate that the initial attempt failed because of the date protection. However, no SAMS:Disk message will be issued unless the subsequent “purge” attempt also fails.

CNVATCYLn

This sysparm controls whether or not the data set's original allocation type (tracks or cylinders) will be used during VSAM restores. The allocation type might change when, for example, the device type changes from the original archived device, or restore parameters (CISZ, IMBED/NOIMBED, SPAN/NOSPAN or RECSZ) alter the data set's original archived values.

The default value of this sysparm is N, meaning no change will be made to the allocation type. Specifying a value of Y allows SAMS:Disk to determine whether or not the allocation type needs to be altered. If the calculated CA-size is less than a cylinder, the allocation will be made in tracks; if equal to a cylinder, the allocation will be made in cylinders.

DARCHIEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DARCHIEX user exit. Turn to the user exit description for *DARCHIEX* on page [228](#) for more information.

DARCKEEXxxx

Specify for "xxx" the number of days a deferred archive request is to be kept in the ARCHCMDS file. This number applies only after the archive has been completed. This allows the results to be inspected by the TSS LISTREQ command. The default is 005 days.

DASDPMAXnnn

Specify a 1- to 3-digit value indicating the total number of pools that may be defined in the DASDPOOL member of parmlib. This value is used to determine memory requirements before building a table with the number of pools and volumes available for non-VSAM volume pooling. See also sysparm DASDVMAX. The default value is 60.

DASDVMAXnnnn

Specify a 1- to 4-digit value indicating the total number of volumes that may be included in defined DASD pools (sum of volumes in all different pools). This also represents the maximum number of volumes that may be specified for a single pool. The default value is 100.

DATECUTO75

This sysparm instructs SAMS:Disk on how to treat 2 digit year values entered from command input parameters or sysparms. Its purpose is to provide the user some flexibility in testing or implementing Year 2000 strategies. The default value is 75, indicating that any 2 digit year value of 75 or less will be considered as 20YY, these dates will be stored as MM/DD/20YY. If the "YY" portion of the date is greater than 75, the dates will be stored as MM/DD/19YY. You may specify any numeric value ranging from 00 to 99.

This allows the use of 2 digits for input to continue, or you may use 4 digit notations as desired. Examples with default value of 75:

- EXPDT=01/01/00 or 00001 will be saved as 2000.001
- EXPDT=01/01/75 or 75001 will be saved as 2075.001
- EXPDT=01/01/76 or 76001 will be saved as 1976.001
- EXPDT=01/01/99 or 99001 will be saved as 1999.001

Caution: There is no ‘special date’ exclusion on inputs. Except for special testing needs, DO NOT set DATECUTO to values of 97, 98, or 99.

For related information, please see “Year 2000 Considerations” on page 9 in the User’s Guide.

DBDSNCFDx

Specify for “x” the index level of the data set name that is to be used as a user code. The default value is 1. This sysparm is activated only if sysparm DBDSNINX has been specified with a value of Y. This sysparm is used in DASD space billing.

DBDSNINXn

Specify a value of Y to indicate that the user code is an index level from the data set name. The default value is N, indicating that the user code is found by sysparms DSCBUSRD and USRCDLEN. This sysparm is used in DASD space billing.

DCAVGCMPhn

Specify for this sysparm the average amount of compression realized when using data compression. To determine this value, run archive or backup with sysparm DCRTSTAT specified with a value of S or D. This value, given as a percent, will be used to estimate the size of a data set while allowing for compression. As a result, a better prediction can be made regarding how many feet of tape or F6470 cartridge will be needed. The default value is 40, or 40 percent average compression.

DCCMPEX0ADSUT371

This sysparm names the module to call for data compression technique number 0. For more information, see the topic “*Data Compression/Decompression*” on page 448 in this manual. Technique number 0 is reserved for SAMS:Disk.

DCCMPEXnnnnnnnnnn

These sysparms name the modules to call for data compression technique numbers 2 - F. Technique numbers 2 through 7 are reserved for future SAMS:Disk-supplied data compression modules. Technique numbers 8 through F are for user-supplied data compression modules. Turn to the user exit description for *DCCMPEXn* on page 229 for more information.

DCCOMPTCn

Specify the technique number to be used for data compression during Archive/Backup/Vbackup processing. This sysparm setting is used when sysparm DCDATACP is specified with a value of Y.

When specified, the default technique number 0 is used for data compression. For more information about how compression is done with this technique, see the topic *"Data Compression/Decompression"* on page 448 in this manual.

Technique number 2 is like technique 0, though not as efficient. However, it requires less CPU overhead than technique 0. Consider using technique number 2 when job performance is more important than compression efficiency.

Technique numbers 0 through 7 are reserved for SAMS:Disk-supplied data compression modules. Technique numbers 8 through F are for user-supplied data compression modules.

For volume-level backup functions, only techniques 0 and 2 are valid. If data compression is turned on (sysparm DCDATACP specified with a value of Y) and a different compression technique is specified, technique 2 will be used.

DCDATACPn

The default value of this sysparm is N, which causes SAMS:Disk to not compress data. Specify a value of Y to activate data compression. For more information, see *"Activating Data Compression"* on page 58 in this manual.

Note: This sysparm is overridden by specification of sysparms DCEXCTBL and DCINCTBL. That is, regardless of the setting of this sysparm, if a data set is in the data compression exclusion table in parmlib, it will not be compressed. If a data set is in the data compression inclusion table, it will be compressed, even if this sysparm is specified with a value of N. See the topic *"Data Compression/Decompression"* on page 448 in this manual.

DCDCPEX0ADSUT372

This sysparm names the module to call for data decompression technique number 0. For more information see the topic *"Data Compression/Decompression"* on page 448 in this manual. Technique number 0 is reserved for SAMS:Disk.

DCDCPEXnmmmmmmmmmm

These sysparms name the modules to call for data decompression technique numbers 2 through F. Technique numbers 2 through 7 are reserved for future SAMS:Disk-supplied data decompression modules. Technique numbers 8 through F are for user-supplied data decompression modules. Turn to the user exit description for *DCDCPEXn* on page 230 for more information.

DCDSNDEXmmmmmmmmmm

Specify this sysparm to provide the module name to be used for screening data sets for data compression. Turn to the user exit description for *DCDSNDEX* on page 230

for more information on this user exit. Turn to page 448 for more information about the data compression/decompression algorithm. Also see “*Activating Data Compression*” on page 58 if you are using data compression technique 1.

DCEXCTBLxxxxxxxx

Specify the SAMS:Disk parmlib member name for “xxxxxxxx” that contains the list of data set or pattern names not to be compressed during the archive/backup. For related information, see “*Activating Data Compression*” on page 58 in this manual.

DCINCTBLxxxxxxxx

Specify the SAMS:Disk parmlib member name for “xxxxxxxx” that contains the list of data set or pattern names to be compressed during the archive/backup.

DCRTSTATn

The default value of N for this sysparm suppresses the informational messages about data compression. Specify a value of S to see archive volume compression totals or D to see individual data set compression totals and archive volume compression totals. See message numbers 2805 and 2806, beginning on page 211 in the *Message Manual*, for format.

DEFARCEXmmmmmmmm

This sysparm causes SAMS:Disk to call a user exit. Specify for “mmmmmmmm” the name of a user-supplied exit module that is to be given control during dispose processing of deferred (queued) archive requests. Turn to the user exit description for *DEFARCEX* on page 231 for more information.

DELETEOKn

The default value of this sysparm is N, which causes SAMS:Disk to reject the DSCL DELETE command with message 2663. To activate the DSCL DELETE command, specify a value of Y.

DERASEEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DERASEEX user exit. Turn to the user exit description for *DERASEEX* on page 233 for more information.

DFLTPOOLxxxxxxxx

Specify an 8-character name to be used as the name of the default DASD pool for data set allocation. This name specifies the name of a pool defined in the DASDPOOL member of the parmlib data set. For syntax information, see “*DASD Pool List*” on page 550 in this manual. If this sysparm is not specified or is specified as blanks, SAMS:Disk will not use a default pool during data set allocation. Note that the pool used in restore and recover applies to non-VSAM data sets only.

DIAG0001n

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAG0002n

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGALOCn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGAUTHn

Normally the SAMS:Disk Security Interfaces print messages only for exceptions to normal processing. Specify this sysparm with a value of Y to have the SAMS:Disk Security Interfaces print additional messages regarding actions and key decisions. The default value of N causes SAMS:Disk to not print these messages. Because this is a diagnostic feature, the content and format of the messages may or may not be documented.

DIAGDCTCn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGENQDn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGENQFn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGENQVn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGEXCPn

This sysparm governs printing of special error diagnostic messages during EXCP processing. The default value is N, which suppresses the additional messages. If

specified with a Y, SAMS:Disk will include the IOB control block, the channel program and the data transmitted or received by the CCW, with any I/O error message returned from the EXCP access method.

DIAGLSTRn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGNOSEn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGRECDn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGSHRKn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGSTAE n

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGSTMPn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGTHRSn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DIAGVSRLn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

DINXUFEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DINXUFEX user exit. Turn to the user exit description for *DINXUFEX* on page 234 for more information.

DISKPROTy

This sysparm provides password protection for archive data sets. The default value of this sysparm is Y, which causes SAMS:Disk to password-protect, but not generate passwords for, archive data sets on disk and bypass the normal security checks when opening them. SAMS:Disk password-protects these data sets by setting the password indicator DS1IND10 bit in the format-1 DSCB. The password indicator prevents access to archive data sets for volumes.

When this sysparm is specified with a value of N, SAMS:Disk does not set the password indicator.

For password-protection of archive tapes, see sysparm *TAPEPROT* on page 190 in this manual.

DLVOLMSGy

The default value for this sysparm is Y, which causes a message to be issued when processing begins for a given volume using DSCL. Specify a value of N to suppress this message.

DRESARCHy

The default value of Y causes the volsers that a data set is archived on to be displayed in message 2401 during a deferred restore request. To determine the volsers that the data set is archived on, the ARCHVOLS subfile must be accessed for INPUT authority.

Specify this sysparm with a value of N to prevent display of the volsers that the data set is archived on and to prevent access to the ARCVHOLS subfile.

DRESKEEPxxx

Specify for “xxx” the number of days a deferred restore request is to be kept in the RESTCMDS file. This number applies only after the restore has been completed. This allows the results to be inspected by the TSS LISTREQ command. The default is 5 days.

DRESTOEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DRESTOEX user exit. Turn to the user exit description for *DRESTOEX* on page 234 for more information.

DSBALTRTn

SAMS:Disk DASD space billing uses a decimal-place unit for accumulating charges when this sysparm is specified with the default value of N. If your installa-

tion requires that charges for DASD space be expressed in a different unit, such as for Japanese yen or Italian lire, specify this sysparm with a value of 1. This changes the billing rates assigned in the DSBRATEC member from a dollars and cents type value to a unit of currency without decimal places.

If this feature is used, specifying “993.25” as an entry in the DSBRATEC member would indicate that either 3.25 lire or 3.25 yen are to be charged for each track or kilobyte day for all volumes assigned to category code 99. The amounts and totals shown on billing reports will be shown as full numbers with no decimal places. The rate value shown on the billing reports will be shown as 5 digits with 2 decimal places.

Note that this alternate rate specification may also be used for other forms of currency as appropriate.

The only valid values for this sysparm are N and 1.

DSBCATSZxxxxx

This sysparm is used in DASD space billing to specify the buffer size for the category table. The default value is 00500.

DSBSELVSn

SAMS:Disk provides the ability to bill at the data set level for both VSAM and non-VSAM. Specify this sysparm with a value of Y to select VSAM data spaces and unique clusters within the non-VSAM billing runs. The default of N causes SAMS:Disk to not to select any VSAM spaces during non-VSAM selection, on the assumption that data set (cluster) level VSAM billing will be done also.

DSBTRKSZt

Specify a value of 'M' to use the maximum data block in determining the kilo-byte days for a data set for billing purposes. The default value of 'T' uses the track capacity value.

DSBUNITSxxx

This sysparm is used in DASD space billing to specify the units used for computing billing cost.

K/B - kilobyte days (default value)

TRK - track days

DSBUSREXmmmmmmmmmm

This sysparm causes SAMS:Disk to call a user exit. Specify for “mmmmmmmmmm” the module name to be given control on every DASD billing record insert or update. The exit is passed the function type and the billing record. Turn to the user exit description for *DSBUSREX* on page [235](#) for more information.

DSBVOLCTnnnn

This sysparm is used in DASD Space Billing to indicate the maximum number of volumes to be processed in the Extend report. The default is 300. If SAMS:Disk issues a message indicating this value is too small, specify a value sufficiently large to allow for some growth. The maximum value allowed is 9999.

DSCBCVEXmmmmmmmm

Specify this sysparm with the module name to be invoked when SAMS:Disk maintained fields must be moved to new offsets within the format-1 DSCB. This user exit is used to move the SAMS:Disk maintained fields during any processing which allocates a user data set on DASD. Turn to the user exit description for *DSCBCVEX* on page [236](#) for more information.

DSCBJBNMxxx

The displacement of the jobname in the DSCB is expressed as a three-digit number. If this field is not present, zeroes should be specified. The default is 062.

SAMS:Disk maintains this jobname in the OPEN SVC. It can be one of the following four values as determined in the OPEN SVC.

1. Creating jobname
2. Last modified jobname
3. Last using jobname
4. Accounting information

DSCBLMODnnnp

The displacement of the last modification date in the DSCB is expressed as a three-digit number. If the date is carried in packed instead of binary format, the letter “p” should follow. If this date is not present in the DSCB, the “nnn” field should be 000. The default displacement is 070.

DSCBLUSDnnnp

The displacement of the last used date in the DSCB is expressed as a three-digit number. If the date is carried in packed instead of binary format, the letter “p” should follow. If this date is not present in the DSCB, the “nnn” field should be 000. The default displacement is 075.

Note: This is the SU 60 compatible offset. Early SAMS:Disk users who have not converted to the SU 60 compatible location (that is, are still using offset 045) must specify a value of 045.

DSCBOPCDnnnh

The displacement of the data set open count is indicated as a three-digit number. If this value is carried as a halfword instead of a fullword, the letter “h” should be

placed after the number. If an open count is not maintained in the DSCB, zeros should be entered. The default displacement is 073.

DSCBSVMDnnn

The SAMS:Disk Open SVC maintains the SVC mode field in the format-1 DSCB. This sysparm indicates the displacement of this field in the format-1 DSCB. The default offset value is 103.

DSCBUSRDxxx

This sysparm is used in DASD space billing, and is the displacement of the user code in the DSCB. The default value is 000.

DSCLMSGSy

This sysparm controls the issuance and suppression of “broken” data set error messages encountered during DSCL SCAN REALVOLS and SCAN CATALOG processing. When this sysparm is specified with the default value of Y, SAMS:Disk is allowed to issue the messages. When this sysparm is specified with a value of N, SAMS:Disk suppress the “broken” data set messages, without affecting the job step return code. Following is a list of the messages that may be issued or suppressed depending on the setting of DSCLMSGs.

2505, 2664, 3262, 3273, 3316, 3397, 3435, 3436, 3437, 3438, 3439, 3440 and 3467.

DSCLOWRCn

This sysparm provides the option of reducing the return code from 4 to 0 when data sets are not selected for processing. When this sysparm is specified with a value of Y, SAMS:Disk will lower the return code in certain circumstances. For details on how to customize this facility, turn to the topic *“User-Specified Condition Codes”* on page [87](#) in this manual.

DSCLSCExmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DSCLSCEx user exit. Turn to the user exit description for *DSCLSCEx* on page [237](#) for more information.

DSDISPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DSDISPEX user exit. The exit module is passed a format-1 DSCB and the TASC field as defined by the retention control VOLSPEC table. Turn to the user exit description for *DSDISPEX* on page [239](#) for more information.

DSNDELEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DSNDELEX user exit. Turn to the user exit description for *DSNDELEX* on page [240](#) for more information.

DSNDLPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the DSNDLPEX user exit. Turn to the user exit description for *DSNDLPEX* on page [241](#) for more information.

DSNDUPVLn

This sysparm is used during IXMAINT, DSCL, and RECOVER processing. IXMAINT processing uses it when the COPIES= parameter is specified. DSCL selection processing uses it when information from the files data set (that is, ARCDATE) is needed. RECOVER processing uses it in the same way.

When this sysparm is specified with the default value of N, SAMS:Disk processing considers DSNINDEX records for data sets with the same name that reside on different volumes to be different data sets and handles them separately for version checking, and retrieval of files data set information.

When this sysparm is specified with a value of Y, SAMS:Disk processing considers them to be the same data set. This allows SAMS:Disk to properly process data sets as they are moved from volume to volume.

See the description of sysparm RESRCAT on page [174](#) for related information.

DSNSELREn

Specifying a value of Y to this sysparm will require the user to code a DSNAMES= parameter on all SELECT statements. The default value of N is equivalent to coding DSNAMES=/ on a SELECT statement which causes SAMS:Disk to attempt to process ALL data sets selected by the SELECT command. This can allow unwanted processing to occur. For example, if a user unintentionally runs:

```
SCAN REALVOLS
SELECT CRITERIA=(DSORG,EQ,VSAM)
DELETE
```

Note: For other related sysparms, refer to the descriptions of SELECTRE (page [184](#) in this manual) and VOLSELRE (page [198](#) in this manual).

DSUTILCFn

The one-digit number designates which index level within the data set name is to be used as the control field for the Data Set Utilization report. The default value is 0, indicating no page breaks on index level change.

DSUTILSQd

The letter N may be specified to designate that the Data Set Utilization report should appear in data set name sequence. The default value is D for date of last use sequence.

DYNCUNITxxxxxxx

Specify for “xxxxxxx” the 1- to 8-character unit name for the F6470 device to be used for dynamic allocation of SAMS:Disk archive cartridges needed for the archive, backup, restore, recover or merge functions. The default unit name is “F6470”. A specific unit address may also be supplied to force all of these functions to use a single F6470 device. For additional information, see the sysparm description for *ARESUNIT* on page 130 in this manual.

Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used. For instance, if you want to simulate other devices, change this sysparm to one of the following values:

Table 4-3. List of Possible Value for DYNCUNIT

Value	Device Simulated	Capacity
F64700	F6470 Cartridge	200 MB

For related information, please turn to the topic *Simulate Mode Considerations* on page 331 of the *User’s Guide*.

DYNEXPDTeeeeeeee

If the //ARCHIVE0 or //ARCHIVEC dd statements for the Archive/Retain functions, or the //ARCHIVE1-5 or //ARCHIVC1-5 dd statements for the Merge function, must be dynamically allocated, this sysparm provides the equivalent of specifying either the EXPDT or the RETPD value in the JCL. For the “eeeeeeee” value, specify either Eyyddd, Eyyyyddd or Rddddd, where E indicates a Julian date follows, and R indicates the following 5 digits are a retention period. A value of E00000 is not valid. The default value is E99365, which has a specific meaning to SAMS:Disk. See “*Assigning Tape Expiration Dates*” on page 42 for applicable rules.

DYNTUNITxxxxxxx

Specify for “xxxxxxx” the 1- to 8-character unit name for the tape devices to be used for dynamic allocation of SAMS:Disk archive tapes needed for the archive, backup, restore, recover or merge functions. The default unit name is TAPE. A specific unit address may also be supplied to force all of these functions to use a single tape device. For additional information, see the sysparm description for *ARESUNIT* on page 130 in this manual.

Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used. For instance, if you want to simulate other devices, change this sysparm to one of the following values:

Table 4-4. List of Possible Values for DYNnUNIT

Value	Device Simulated
01600	F619 at 1600 BPI
06250	F619 at 6250 BPI

For related information, please turn to the topic *Simulate Mode Considerations* on page [331](#) of the *User's Guide*.

DYNUKEEPn

This sysparm is used with dynamic allocation in all of the archive, backup, restore, recover and merge functions to determine whether or not the tape device should be released each time a new tape volume is needed. It applies to both input and output tapes. If this sysparm is specified with a value of N, the default value, the device is freed at completion of each archvols tape. If the sysparm is specified with a value of Y, the same device is kept for the entire job step.

Keeping the same device may aid the operations staff in knowing where tape mounts will be needed, but it may also increase run time since the next tape cannot be mounted until the prior one is rewound and dismounted.

DYNVREQDn

When dynamic allocation is being used, this sysparm controls whether mount requests for output archive volumes will occur at allocation time or at OPEN time. The default value of N means the archive volumes are mounted at OPEN time. If this sysparm is specified with a value of Y (and sysparm DYNUKEEP is also specified with a value of N), archive volumes are mounted at allocation time.

DYNnUNITxxxxxxx

Specify a 1- to 8-character unit name for special device type DYN1 (2 or 3). This unit name is used for dynamic allocation of archive volumes during archive, backup, restore, recover, retain and merge functions, whenever DYN1 (2 or 3) has been specified as the device type to be used via one of the sysparms ARC0TYPE, ARCCTYPE, MERPnTYP, or MERCnTYP. When DYN1 (2 or 3) is used, a flag is set in the ARCHVOLS record and the unit name specified in DYN1UNIT (DYN2UNIT or DYN3UNIT) is used for all functions relating to that archive volume. A specific unit address or an esoteric name that maps only to the special devices is recommended as a value for these sysparms.

Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used. For instance, if you want to simulate other devices, change this sysparm to one of the following values:

Table 4-5. List of Possible Value for DYNnUNIT

Value	Device Simulated	Capacity
01600	F619 at 1600 BPI	
06250	F619 at 6250 BPI	
F64700	F6470 Cartridge	200 MB

For related information, please turn to the topic *Simulate Mode Considerations* on page [331](#) of the *User's Guide*.

ENQADDTLn

If this sysparm is set to a value of Y and the USERSENQ user exit is supplied with a value of DMSNOENQ, an additional enqueue will be performed to determine if a data set is in use. If so, a message will be issued stating which data sets were in use at the time of the backup. The default value is N.

ENQDSSIMn

This sysparm controls data set ENQUEUEES issued during PDS Compress processing in simulate mode. The default value of this sysparm is N, which causes SAMS:Disk to bypass data set ENQUEUEES in simulate mode. This eliminates data set contention and improves performance, but the report will include data sets that would not have been selected and processed in a live run because they were in use by another task.

If you need a more accurate report from the simulate run, specify this sysparm with a value of Y. SAMS:Disk will perform data set ENQUEUEES during the simulate run, and thus select and report the same data sets that would have been selected in a live run.

ENQGDGBYn

This sysparm affects the enqueueing on GDG data sets. It allows you to optionally bypass the enqueue on the base name of a GDG.

The default value for this sysparm is N. Per Fujitsu integrity conventions for generation data groups, SAMS:Disk enqueues on the base name before enqueueing on the full data set name. Therefore the enqueue on the base name, whether by SAMS:Disk or another application, can prevent other jobs from accessing any generation within the group; for example, SAMS:Disk will bypass all generations within the group until the other task releases (dequeues) the base name.

If you specify this sysparm with a value of Y, SAMS:Disk will bypass the enqueue on the base name of the GDG data set and process individual generations, based upon their enqueue status. Since Fujitsu conventions are not being used, unpredictable integrity exposures may arise for which SAMS:Disk is no longer responsible.

ENQMSGESn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

ENQSTPOKn

When this sysparm is specified with the default value of N, SAMS:Disk will not process data sets allocated in the job step that is currently running. Enqueues for these data sets normally fail. This is to prevent SAMS:Disk from processing (that is, archive, delete, etc.) the data sets it has in use which may sometimes cause unpredictable results.

Specifying this sysparm with a value of Y will allow SAMS:Disk to process data sets in the job step that is currently running. Problems mentioned above may occur, however. Other errors may also occur. For example, you may receive SVC99 errors if you attempt to restore a data set which is also allocated to the currently running step.

This sysparm does not apply to auto-restore processing.

ENQVOLWTnnn

This sysparm controls the length of time between issuance of WTOR message 3017, which tells the operator how long SAMS:Disk has been waiting for a tape volume that is being used by another job. The default value is 8, for 8 minutes. You may specify a value from 1 to 999 minutes.

ENQWAITDn

This sysparm controls the Enqueue action when SAMS:Disk is processing its selection criteria. The default value of N allows SAMS:Disk to process non-VSAM and VSAM data sets when enqueues can be immediately obtained.

Other possible values are:

- V** — wait on enqueues for VSAM data sets only
- Y** — wait on enqueues for non-VSAM data sets only
- B** — wait on enqueues for both VSAM and non-VSAM data sets

EXPARSOPy

The default value for this sysparm is Y, which causes the change bit to be reset (turned off) after an explicit batch archive that does not scratch the data set. Specify a value of N to change the default to no reset.

EXPASCOPs

Specify a value of N to change the option to no scratch in explicit batch archive. The default is S, for scratch.

EXPDTUSEn

When data sets are archived, SAMS:Disk assigns an expiration date from either the SAMS:Disk command or from a default value set by a sysparm. If you wish to use the expiration date from the data set's VTOC entry (or its catalog entry for VSAM), if one exists, specify this sysparm with a value of Y. The default value is N, which disables this feature.

If Y is specified and the expiration date from the VTOC entry (catalog entry for VSAM) is less than the calculated default value, the default value will still be used.

Warning: This option applies to the expiration date being assigned to each DSNINDEX record, and does not directly affect the expiration date of associated ARCHVOLS records. All archive functions should use an expiration date of 99365 for the archive volumes if this sysparm option is used. Failure to do so may result in archive volumes expiring before the data sets contained on them are expired. Supply this expiration date in the JCL or by using sysparm DYNEXPDT.

FASTCOMPY

The default value of Y causes SAMS:Disk to compress PDSs using a newer faster method. Specify a value of N only if directed by the Technical Support Center.

FILESCNT12

This sysparm controls how many subfiles can be contained in the files data set. The default value is 12.

FILES PAC1500

This sysparm is active only when the files data set logging feature of SAMS:Disk has been activated (see sysparm FILOGNAM). This sysparm controls the number of blocks allocated when the files log data sets are created. The default value is 1500, causing 1500 blocks to be allocated. You may specify this sysparm with any 4-digit numerical value.

FILEUNITnnnnnnnn

This sysparm is active only when the files data set logging feature of SAMS:Disk has been activated (see sysparm FILOGNAM). This sysparm controls the type of unit allocated when the files log data sets are created. The default unit name is SYSALLDA. You may specify any valid unit type for your installation.

FILOGNAMxxxxxxxxxxxxxxxxxxxxxxxx

Specify this sysparm to activate the files data set logging feature of SAMS:Disk. For the value of this sysparm, specify the base name (high-level node) to be used to create the files log data set name. The base name can be 1 to 25 characters. When this sysparm is specified, SAMS:Disk will dynamically allocate data sets to log activity to the files data set.

The default is blanks, which disables logging for the files data set.

FLYCOUNT2

SAMS:Disk prints report separator pages on the SYSPRINT dd statement for all SAMS:Disk executions. It appears as the first page(s) printed before SAMS:Disk-generated output. We refer to the separator pages as “fly sheets”. They provide two benefits for SAMS:Disk users; first, as a separator page for different report steps, and second, to provide useful information about the environment in which SAMS:Disk is running.

This sysparm controls the printing of SAMS:Disk fly sheets. It specifies the number of fly sheets to be printed. To suppress their printing, specify a value of 0 for this sysparm. To obtain two fly sheets before printed output, specify a value of 2, the default value. The value must be a single numeric digit.

FORCPOOLnnnnnnnn

When tape or disk archive pools are being used, the pool names associated with the archive tapes and disk volumes are specified in the POOLDEFS member of the parmlib data set. A pool name supplied in the “nnnnnnnn” field of this sysparm will override the normal definitions, forcing all archive volumes to be taken and returned to the named pool. This sysparm is intended for use in special testing or recovery environments only.

FMSSAFEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the FMSSAFEX user exit. Turn to the user exit description for *FMSSAFEX* on page [243](#) for more information.

FMSSPREXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the FMSSPREX user exit. Turn to the user exit description for *FMSSPREX* on page [243](#) for more information.

GDGCATEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the GDGCATEX user exit. Turn to the user exit description for *GDGCATEX* on page [243](#) for more information.

GDGDSNCT00000200

SAMS:Disk builds an incore table of GDG data set names when processing functions that need to determine the relative generation numbers of GDG data sets. This table allows SAMS:Disk to keep track of the base generation number of a GDG data set. By keeping this table, SAMS:Disk minimizes catalog access during scans of DASD volumes, thus improving performance.

This sysparm specifies the number of GDG base names that are to be kept in the incore table (the default is 200). If more GDG names are processed than there are entries in the table, old entries will be deleted to make room for new ones. They are

deleted in a least- recently used fashion. Each entry in the table requires about 50 bytes of storage.

HONORTTRn

This sysparm controls whether SAMS:Disk Restore or Move/Copy processing will honor the TTR value of records in data sets of DA, PS and unknown organization types.

The default value for this sysparm is N. This causes SAMS:Disk to honor the TTR for all data sets with DSORG=DA and a RECFM=U or V. At archive time, any other DA, PS or unknown DSORG data set in which an empty track is encountered prior to the end-of-file is also marked as requiring this processing at restore time. For other attribute combinations of DA, PS and unknown DSORG data sets, SAMS:Disk will copy the records in correct order but, depending on the data and device characteristics, the records may be moved to different relative track and record locations, per standard Fujitsu conventions.

Note: For PS data sets with RECFM=FBS, this sysparm should be set to N.

If you specify this sysparm with a value of Y, Restore and Move/Copy processing will preserve the TTR value for data sets of these organizations (DSORGs), regardless of what the record formats (RECFMs) are. Specifying a value of Y for this sysparm is intended only for those exception cases in which the data sets do not follow/allow conventional access techniques for the attributes that are indicated.

ICHEKAEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ICHEKAEX user exit. Turn to the user exit description for *ICHEKAEX* on page [244](#) for more information.

ICHEKPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the ICHEKPEX user exit. Turn to the user exit description for *ICHEKPEX* on page [245](#) for more information.

IDRCUSEDx

This sysparm indicates whether or not EDRC is activated when creating output tapes. The values that can be assigned are blank, Y, or N. When the sysparm is specified with the default value of blank, it means that whatever system setting exists as a result of the installation default and/or JCL parameter specification should be used. When the sysparm is specified with a value of Y, it means that the compaction feature (EDRC) should be activated if the capability is available. When the sysparm is specified with a value of N, it means that the compaction feature should not be activated when creating output tapes.

IGNLSTARy

With this sysparm, SAMS:Disk determines if a non-VSAM Data set is empty for the purposes of archiving, backing up, or Move/Copy processing.

When SAMS:Disk checks if a non-VSAM data set is empty, it applies the following tests:

- the DS1LSTAR value in the format-1 DSCB is zero
- the DS1TRBAL value in the format-1 DSCB is non-zero
- the DS1VOLSQ value in the format-1 DSCB is one

The last test is based on the above sysparm setting. If the sysparm is set with a value of Y, the next test is also applied. The default value is Y.

bit 0 of the DS1DSIND field in the format-1 DSCB is zero

If the value of this sysparm is N, this last check is not done.

If the data set meets all of these criteria, it is backed up or copied as empty. If the data set does not meet these criteria, it is not considered empty and is processed accordingly.

Some data sets meeting all but the last criterion could have been opened for output. So, data may exist if the program that wrote to the data set did not update the DS1LSTAR field. To capture data sets that may not be truly empty, specify this sysparm with a value of Y.

This sysparm does not apply to direct access data sets with variable or undefined record formats. These data sets are never considered empty and are always backed up or copied until an end-of-file is detected.

IOERRLIMnnnnn

This sysparm determines the number of I/O errors allowed during a VBACKUP run before the volume should be bypassed. The default for this sysparm is 0, indicating that a volume is to be bypassed if any I/O error is encountered. Specify a 1- to 5-digit value to indicate the number of allowable I/O errors.

During VBACKUP, the number of I/O errors as specified in this sysparm up to a maximum of 100, are saved. For example, if a user specifies 150 for the IOERRLIM sysparm, 100 of the I/O errors will be saved. SAMS:Disk will continue processing the volume until more than 150 I/O errors have been encountered. SAMS:Disk saves the CCHH value for the track in error, along with two sense bytes indicating the type of error. The I/O errors detected are written to the archive tape as one record.

IOMAXRECnnnnnn

This sysparm specifies the maximum logical record size for data sets processed by SAMS:Disk. The default value is 65,000 bytes. The maximum value for this sysparm is 999,999 bytes. To save real storage, this sysparm should be set to the lowest value possible that still allows normal SAMS:Disk processing.

IOTRACKSxx

This sysparm controls the number of tracks that SAMS:Disk can read (for input processing) or write (for output) with a single I/O request. Since a buffer is required to hold the data for each I/O, this sysparm also establishes the buffer size. To allow I/O and CPU processing to be overlapped, SAMS:Disk always acquires two buffers for each data set that is opened. As should be evident, the value of this parameter has a very dramatic affect upon the amount of memory that will be required.

The default value is 30, which allows 30 tracks to be read or written with a single I/O. If the value specified is larger than the number of tracks per cylinder for the device being processed, SAMS:Disk sets it to the tracks per cylinder value; that is, 15 for a F6425, 30 for a F493, etc.

Processing multiple tracks per I/O reduces the total I/O activity against the channels, control units and devices, and generally “improves” performance. However, memory requirements, paging activity, and many other factors affect performance and must also be considered when selecting and testing a value for this parameter.

ISAMRECOc

This sysparm indicates what SAMS:Disk is to do after encountering an abend during processing of an ISAM data set. When an abnormal condition in an ISAM data set causes ISAM access methods to terminate with a system abend, SAMS:Disk intercepts the system abend, prints messages, and attempts to continue processing. The abend may have left ISAM system control blocks in a status that does not allow further ISAM processing. If SAMS:Disk attempts to process another ISAM data set, unpredictable results can occur. This sysparm indicates what SAMS:Disk is to do if an attempt is made to process another ISAM data set after an ISAM abend has previously occurred. The valid values are:

C	Continue and attempt to process additional ISAM data sets. Processing another ISAM data set may cause SAMS:Disk to abend (default value).
A	Abend SAMS:Disk if another ISAM data set is selected for processing.
B	Bypass processing for any additional ISAM data sets selected for processing.

IXCTLGEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the IXCTLGEX user exit. Turn to the user exit description for *IXCTLGEX* on page 246 for more information.

IXMCLEANy

This sysparm indicates if the SAMS:Disk archive index maintenance utility (IXMAINT JCL procedure) is to remove invalid records from the DSNINDEX and ARCHVOLS subfiles during processing. ARCHVOLS records must relate to existing DSNINDEX records, and DSNINDEX records must relate to existing ARCHVOLS records. Invalid records are those for which corresponding records do not exist in the opposite file.

The default value is Y, which indicates that invalid records are to be removed when found. A value of N indicates that records are to remain in the files.

IXMDUPDTn

Normally, SAMS:Disk archive index maintenance does not consider data sets archived on the same date to be different versions. For this reason the COPIES= parameter will not dispose of them as desired when this sysparm is specified with a value of N, the default value. If data sets with duplicate data set names archived on the same date are to be considered as different versions, specify this sysparm with a value of Y.

IXMREVEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the IXMREVEX user exit. Turn to the user exit description for *IXMREVEX* on page 248 for more information.

IXMVOLONn

The default value of this sysparm is N, which causes SAMS:Disk index maintenance (DSNDELETE, VOLDELETE, PURGE) and merge processing to uncatalog obsolete catalog entries (see sysparm UNCATPSU on page 195 for related information) for data sets on offline volumes. Specify a value of Y to not uncatalog data sets on offline volumes.

IXUSRNOTn

This sysparm indicates if the SAMS:Disk archive index maintenance utility (IXMAINT JCL procedure) is to send a TSS message to a userid if only one copy of a data set is left in the files data set and that data set is uncataloged.

The default value is N, which indicates that no message is to be sent.

A value of Y indicates that a warning TSS message is to be sent. The TSS message warns that there is only one copy of the data set left. It is sent to the userid that either archived or backed up the data set, or if the expiration date for the data set in the files data set has been updated, to the userid that last did the update.

LISTDMEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the LISTDMEX user exit. Turn to the user exit description for *LISTDMEX* on page 249 for more information.

LISTDVEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the LISTDVEX user exit. Turn to the user exit description for *LISTDVEX* on page 249 for more information.

LISTREXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the LISTREEX user exit. Turn to the user exit description for *LISTREEX* on page 250 for more information.

LISTSORTn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

LISTVTOCn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

MERCnNAMoooooooooooooooooooooooooooo

Merge processing can intelligently redistribute archived data sets on the basis of their expiration dates, by the use of the `LIMITS=` parameter. Two limit values cause a 2-way split, three a 3-way split, with a 5-way split being the maximum. Each of these new primary archive data sets may also have a duplicate copy created concurrently (duplexing). The new primary data sets are referenced via `//ARCHIVE1` (2, 3, 4, 5) dd statements, and the copies by `//ARCHIVC1` (2, 3, 4, 5).

If MSP dynamic allocation is used to allocate the //ARCHIVC1 (2, 3, 4, 5) dd statements, these sysparms provide the data set names to be used. The nnn value is either the 1- to 22-character base name, the complete name or the tape GDG index, depending on installation JCL and other sysparm options. The default values are DMS.MERGCOP1 (2, 3, 4, 5) for the duplicate copy data sets created by the merge process.

MERCnTYPtape

If the //ARCHIVC1 (2, 3, 4, 5) dd statement is not provided in the JCL, these sysparms indicates the type of device to be dynamically allocated. It must specify one of the following: TAPE, F6470, DISK, DYN1, DYN2, DYN3 or NULL. TAPE is the default.

MERDSNEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the MERDSNEX user exit. Turn to the user exit description for *MERDSNEX* on page [251](#) for more information.

MERFASTSy

Normally, simulate processing of the merge function does not invoke SAMS:Disk user exits. This is done to improve the performance of processing. To cause simulate processing to invoke the user exits, specify this sysparm with a value of N.

MERPnNAMnnnnnnnnnnnnnnnnnnnnnn

Merge processing can intelligently redistribute archived data sets on the basis of their expiration dates, by the use of the LIMITS= parameter. Two limit values cause a 2-way split, three a 3-way split, with a 5-way split being the maximum. Each of these new primary archive data sets may also have a duplicate copy created concurrently (duplexing). The new primary data sets are referenced via //ARCHIVE1 (2, 3, 4, 5) dd statements, and the copies by //ARCHIVC1 (2, 3, 4, 5).

If MSP dynamic allocation is used to allocate the //ARCHIVE1 (2, 3, 4, 5) dd statements, these sysparms provide the data set names to be used. The nnn value is either the 1- to 22-character base name, the complete name or the tape GDG index, depending on installation JCL and other sysparm options. The default values are DMS.MERGPRI1 (2, 3, 4, 5) for the primary data sets created by the merge process.

MERPnTYPtape

If the //ARCHIVE1 (2, 3, 4, 5) dd statement is not provided in the JCL, these sysparms indicate the type of device to be dynamically allocated. It must specify one of the following: TAPE, F6470, DISK, DYN1, DYN2 or DYN3. TAPE is the default.

MERREPLYn

SAMS: Disk merge processing may be instructed to allow the operator to terminate the merge process normally through operator reply, between any input tape mount. If you wish to activate this feature, specify **Y** for the value of this parameter. The default value is **N**. Review the topic *"Operation Concerns"* on page 328 in the *User's Guide* for more details about this feature.

MERVOLExmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the MERVOLEX user exit. Turn to the user exit description for *MERVOLEX* on page 251 for more information.

MIGBYPAS_n

Specify a Y for this sysparm if you desire to see the reasons why the sequential migrate to tape function has bypassed a data set. The default value of N means that no messages are given for data sets not meeting selection criteria specified.

MIGCATLGy

The default value of Y for this sysparm will cause all uncataloged non-GDG data sets to be cataloged when migrated to tape. Specifying a value of N will cause no change to the catalog.

MIGCREDTn

Specify a value of Y for this sysparm to cause all data sets migrated to tape to have their creation date equal to the creation date the data set was assigned on DASD. The default value of N indicates that SAMS:Disk is to reset all creation dates to the current date.

MIGCTGDGn

Specify a value of Y for this sysparm to cause all uncataloged GDG data sets to be cataloged when migrated to tape. The default value of N will cause no change to the catalog. If a GDG data set is not in the catalog, no attempt is made to catalog it because VSAM catalogs will uncatalog the oldest generation when the index is full. (See NOEMPTY parameter for the DEFINE GENERATIONDATAGROUP command in Fujitsu's Access Method Services manual.)

MIGDISEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the MIGDISEX user exit. Turn to the user exit description for *MIGDISEX* on page 252 for more information.

MIGDSMAXnnnn

This sysparm specifies the maximum number of data sets that may be migrated to a single tape volume. The default value is 9999.

MIGDSORGn

Specify this sysparm with a value of Y if data sets with a null DSORG are to be migrated during sequential migration as if they were physical sequential. The default value is N, which causes SAMS:Disk not to migrate null DSORG data sets. This parameter applies only to the implicit SCAN command.

MIGEXPDTn

Specify a value of Y for this sysparm if the expiration date is to be taken from the JCL. The default value is N, causing SAMS:Disk to use the data set expiration date.

MIGFXPDty

Specify a value of N for this sysparm to cause SAMS:Disk, in sequential migrate, to not add one day to the expiration date of the data set when it is equal to the current date. Tape management systems consider data sets on tape with an expiration date of the current date to be a tape work file and will not log the data set in the tape management catalog.

MIGGDGEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the MIGGDGEX user exit. Turn to the user exit description for *MIGGDGEX* on page [253](#) for more information.

MIGMODELn

This sysparm's default value of N causes model DSCBs to be bypassed during implicit SCAN processing for sequential migration to tape. Specify Y to permit model DSCBs to be migrated.

MIGNOEXPn

This sysparm's default value of N causes expired data sets (non-zero expiration dates less than current date) to be bypassed during implicit SCAN processing for sequential migration to tape. Specify Y to permit expired data sets to be migrated.

MIGNXPDTy

This sysparm's default value of Y allows data sets with zeroes in the expiration date field to be selected during implicit SCAN processing for sequential migration to tape. Specify N to cause these data sets to be bypassed.

MIGPSCOPn

Specify a value of Y to create a duplicate backup tape concurrently with the primary sequential migration tape. The default value is N, for no duplicate.

MIGREXITmmmmmmmmmm

This sysparm allows you to specify the 8-character name of the user-supplied exit module to receive control after a data set has been selected for migration from disk to tape. Turn to the user exit description for *MIGREXIT* on page [254](#) for more information.

MIGRSORTndv

Specify this sysparm to change the selection of sequential migration reports produced. Specifying values of UDV will cause SAMS:Disk to produce migration reports sorted by different fields as describes below:

Table 4-6. Possible Values for sysparm MIGRSORT

Value	Description
U	print unsorted sequential migration report
D	print migration report in dsn/volume sequence
V	print migration report in volume/dsn sequence

Specify N for any report not desired. The default values are NDV.

MIGYRCHGn

Specify a value of Y if you want to have sequential migrate to tape mount a new tape volume when the expiration year changes for the data sets being migrated. The default value of N causes no tape switching to be done.

NOBLKMSGn

Specify a value of Y if you want to see the reason why a data set was not reblocked by the Move/Copy function. The default value of N means no messages are issued explaining why reblocking was not done.

OFFLDASDn

Specifying a value of Y instructs the RESTORE command to issue a DELETE NOSCRATCH catalog action prior to restoring the data set. This catalog action helps prevent abends in cases where the data set being restored has ONLY a catalog entry, or when the target volume is offline (no valid UCB). The default value of N bypasses any catalog action.

Note: This sysparm is not valid for the RECOVER command.

PASSNEWNn

Specify a value of Y if you wish to allow password- indicated non-VSAM user data sets to be renamed. Since SAMS:Disk does not update the password control data set itself, renaming a password-indicated data set can be a security exposure. The default value of N prevents this condition.

PASSWORDn

The default value for this sysparm is N, which causes SAMS:Disk to bypass processing for all password-indicated data sets when they are encountered. Whenever a password-indicated data set is bypassed, a message to that effect will be issued.

Specifying a value of Y allows SAMS:Disk to process password- indicated data sets. The user may also have to use the PASSWORD parameter on the appropriate SAMS:Disk command. Processing of password-indicated data sets is supported in all functions.

If you have the security package SECURE installed, you should also specify sysparm SECUSUPP with a value of Y. You should also consider specifying sysparm PASSNEWN.

If sysparm SECUSUPP is not specified with a value of Y, SAMS:Disk supports processing for password-indicated data sets only in the MSP and SVS environments (SVS requires the installation of a system zap modification and the SAMS:Disk SVC). If the above conditions are not met, the value of N for this parameter will always be assumed.

PDSANMLYmmmmmmmm

This sysparm allows you to specify the 8-character name of a member in the parm-lib data set that contains the PDS Anomaly Code and Options Specifications. Review the topic *"Processing PDSs that Contain Anomalies"* on page 71 for further information regarding SAMS:Disk-provided options to bypass or continue processing partitioned data sets in which SAMS:Disk has detected errors.

PDSENQSHn

The default value for this sysparm is N, which causes SAMS:Disk to obtain an EXCLUSIVE ENQ during PDS Compress processing.

If this sysparm is specified with a value of Y and the USE parameter is specified for the compress command, a SHARED ENQ is maintained for the data set during compression. (Without the USE parameter, a value of Y for this sysparm is ignored.)

Using a SHARED ENQ might be desirable, for example, to recover dead space in TSS PDSs during high use periods, since the SHARED ENQ could be obtained more readily. And because SAMS:Disk also issues the PFD and linkage editor enqueues when using a SHARED ENQ, processing between TSS/PFD and PDS Compress are serialized, preserving the data set's integrity.

Specifying this sysparm with a value of Y has no effect on the RESERVE command parameter for PDS compress processing.

Note: Specifying this sysparm with a value of Y can cause an integrity exposure if a batch job (other than the linkage editor) is updating a PDS that has only a SHARED ENQ.

Also, if a SHARED ENQ is used in an environment in which other batch jobs are using the PDS concurrently, unpredictable results may occur; for example, member not found, S806-04 and S106 abends, and so on. In this case, rerun the batch jobs when compress processing has completed.

PDSEXSIMy

Specify a value of N for this sysparm if the exclusion table should be nullified during simulated PDS compression. The default value of Y indicates that the table is active even in simulate mode.

PDSRRMAXnnnnn

The specified "nnnnn" value determines how many 3-byte TTR entries will be reserved in an internal SAMS:Disk table. The table is used to maintain relative block numbers when processing PDS members with user TTR entries. The default value is 03000. If this value is too small, an error message will indicate to increase its value.

PDSSETBSy

The default value of Y allows SAMS:Disk to set an optimal blocksize for the work file used by the PDS Compress function. Specify a value of N if you want

SAMS:Disk to honor the blocksize supplied on the //COMPWORK DD statement of the COMPRES step.

PDSMAXPR00000

This sysparm is used during PDS Compress to set the maximum number of primary tracks to be calculated for reallocation of the PDS. If the SAMS:Disk calculation for primary tracks generates a number greater than this number, the value of this sysparm will be used unless it is less than the number of tracks used in the PDS. In this case, the actual number of used tracks will be used. If the value of this sysparm is set to zero, no maximum limit is set. This sysparm is in effect only if the user has specified a value other than 0 in the primary space specification of the ALLOCATE= parameter. The default value of this parameter is 00000.

If the PDS being compressed is allocated in cylinders, this value will be converted to a cylinder value. During this calculation it may be necessary to round the value up to a higher value that will fall on a cylinder boundary.

PDSMAXSE99999

This sysparm is used during PDS Compress to set the maximum number of secondary tracks to be calculated for reallocation of the PDS. If the SAMS:Disk calculation for secondary tracks generates a number greater than this number, the value of this sysparm will be used. If the value of this sysparm is set to zero, the secondary extent allocation amount will be set to zero. This sysparm is in effect only if the user has specified a value other than 0 in the secondary space specification of the ALLOCATE= parameter. The default value of this parameter is 99999 tracks.

If the PDS being compressed is allocated in cylinders, this value will be converted to a cylinder value. During this calculation it may be necessary to round the value up to a higher value that will fall on a cylinder boundary.

PDSMINDB00000

This sysparm is used during PDS Compress to set the minimum number of directory blocks to be calculated for reallocation of the PDS. If the SAMS:Disk calculation for directory blocks generates a number of directory blocks to allocate less than this number, the value of this sysparm will be used instead. This sysparm is in effect only if the user has specified a value other than 0 in the directory block specification of the ALLOCATE= parameter. The default value of this parameter is 00000.

PDSMINPR00000

This sysparm is used during PDS Compress to set the minimum number of primary tracks to be calculated for reallocation of the PDS. If the SAMS:Disk calculation for primary tracks generates a number less than this number, the value of this sysparm will be used instead. This sysparm is in effect only if the user has specified a value other than 0 in the primary space specification of the ALLOCATE= parameter. The default value of this parameter is 00000.

If the PDS being compressed is allocated in cylinders, this value will be converted to a cylinder value. During this calculation it may be necessary to round the value up to a higher value that will fall on a cylinder boundary.

PDSMINSE00000

This sysparm is used during PDS Compress to set the minimum number of secondary tracks to be calculated for reallocation of the PDS. If the SAMS:Disk calculation for secondary tracks generates a number less than this number, the value of this sysparm will be used instead. This sysparm is in effect only if the user has specified a value other than 0 in the secondary space specification of the ALLOCATE= parameter. The default value of this sysparm is 00000.

If the PDS being compressed is allocated in cylinders, this value will be converted to a cylinder value. During this calculation it may be necessary to round the value up to a higher value that will fall on a cylinder boundary.

POINTABSy

The default value of Y allows SAMS:Disk to determine if a F6470 is in full function mode or not, and to use POINT TYPE=ABS (high speed search) whenever possible. You may specify N to instruct SAMS:Disk not to use the high speed search facility.

PREALLOCn

The default value for this sysparm is N, which causes SAMS:Disk not to allow restore or recover operations to overwrite preallocated non-VSAM data sets. Specify a value of Y to allow Restore operations to overwrite pre-existing non-VSAM data sets (providing the DSORG of the target data set matches the DSORG of the data set being restored, otherwise the restore will fail). Specify a value of C to allow the overwrite only if the preallocated data set's create date is the same as that of the archived version.

See sysparm *VSPREDEF* on page 208 for restore of VSAM data sets.

In order to allow concurrent SAMS:Disk member-level restore processing and PFD editing of a PDS, SAMS:Disk internally assigns and uses a value of "Y" for this sysparm when the MEMBER= parameter is specified and the ERASE parameter is not specified.

(To protect integrity, the SAMS:Disk auto-restore function internally assigns and uses a value of "N" for this sysparm; that is, an auto-restore will not overlay an existing data set.)

PRIALLOCn

Specify a value of 'Y' to use the original primary allocation amount of the data set for allocation during the PDS Compress, RESTORE, RECOVER, COPY, and MOVE functions. The default value of 'N' uses the consolidated allocation amount unless it would exceed the volume capacity.

PRTSORTFy

The default value of Y causes a footing line showing the sort order to be printed for online reporting. Specify N to not print the sort footing line.

PSUDEVTPn

When recataloging archived data sets to the volume specified in the RECATVOL sysparm, the default is to put an invalid device type in the catalog entry for that data set. This is done on the assumption that the RECATVOL volume is a pseudo-volume, and you want to prevent the operating system from calling for a mount of the pseudo-volume when the data set is referenced. A JCL error results indicating the data set must be restored first. If you desire to keep the original device type of the data set, specify a value of Y for this sysparm. The default value is N.

If you specify a value of Y for this sysparm and SAMS:Disk attempts to recatalog a data set to its original device but this device is offline, sysparm REDEVDEF can be specified to cause SAMS:Disk to use a default device (F6425) rather than issue an error message. See sysparm *REDEVDEF* on page [173](#) for further information.

If you have implemented only the IFG0EX0A (S213 abend) hook of the auto-restore support and wish to direct the restored data sets to a common “staging” volume, you must specify a real online volume for the RECATVOL sysparm and specify this sysparm with a value of R. This causes SAMS:Disk to obtain the correct device type to be placed in the catalog entry.

If you are using the catalog management hook provided by SAMS:Disk for your auto-restores, this sysparm should be left with the default value of N and the RECATVOL sysparm should specify a non-existent (pseudo) volume. The default value for the RECATVOL sysparm is *ARCIVE*.

RAADDAEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RAADDAEX user exit. Turn to the user exit description for *RAADDAEX* on page [254](#) for more information.

RAADDPEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RAADDPEX user exit. Turn to the user exit description for *RAADDPEX* on page [255](#) for more information.

RACFALLOa

This sysparm is functional only if you have installed the SAMS:Disk RACF security interface. It allows you to control what level of authority SAMS:Disk will check for when allocating data sets. The default value of A causes SAMS:Disk to check for RACF ALTER authority before allocating data sets in the Restore, Recover and Move/Copy functions. Before allocating the data set, you may instruct

SAMS:Disk to check for certain authority by specifying this sysparm with a value of:

Table 4-7. Possible Values Available for RACFALLO

Value	Meaning
C	to check for CONTROL authority
U	to check for UPDATE authority
R	to check for READ authority
N	to perform no authorization check

If you have a data set allocation control product or site-written DADSM KBYPRE00 allocation exit that checks for RACF ALTER authority before permitting data set allocations, use the default value for this sysparm. If you have generic profiles that disallow most users access to most usable data set names, trusting the ADSP feature of RACF to provide access to newly allocated data sets, specify this sysparm with a value of N. If you have a DADSM KBYPRE00 exit that checks for other than ALTER authority, specify this sysparm with a value consistent with your exit.

If you do not have a DADSM KBYPRE00 exit and are not a heavy user of the ADSP feature of RACF, specify this sysparm with a value consistent with your data management and security philosophy.

RACFALWZy

Under most versions of Fujitsu operating systems, OPEN, SCRATCH and RE-NAME processing will query RACF for authorization regardless of the setting of the RACF-indicator bit. This feature is called always call. Data sets cataloged in EDF-Catalogs also cause a query of RACF for authorization regardless of the setting of the RACF-indicator bit.

Under some operating systems, data sets not cataloged in EDF-Catalogs will query RACF only if the RACF-indicator bit is on. For non-VSAM data sets, the RACF-indicator bit is the DS1IND40 bit (bit x'40' at offset 93 x'5D') located in the data set's format-1 DSCB.

SAMS:Disk security processing normally queries RACF for authorization regardless of the setting of the RACF-indicator bit. If you do not have the always call feature of the operating system and you do not use EDF-Catalogs, specify this sysparm with a value of N. The default value is Y.

RACFBITSy

When this sysparm is specified with the default value of Y, SAMS:Disk preserves the value of the RACF-indicator bit. Preserving this indicator bit is good security management in all installations that use RACF. For non-VSAM data sets, the

RACF-indicator bit is DS1IND40 (bit x'40' at offset 93 x'5D') in the format-1 DSCB. For VSAM data sets, the RACF-indicator bit is the x'80' bit in the SECF field in the cluster's catalog entry. If you specify this sysparm with a value of N, SAMS:Disk never sets the RACF-indicator bit in the data sets that it allocates. If you do not have a security package, and may need to restore SAMS:Disk backup or archive tapes from sites with RACF, set this sysparm to N to prevent security errors on those restores. To prevent security exposures, installations with RACF discrete profiles should NOT set this sysparm. To do so would cause data sets to become protected by generic profiles instead of their discrete profiles.

RACFBKUPn

If this sysparm is not specified or is specified with a value of N, any SAMS:Disk function that archives or backs up a data set protected by a discrete RACF profile will create a SAMS:Disk profile for the data set.

For some users, the additional profile may be costly both in terms of job performance and DASD space utilization in the RACF profile data set.

Generic profiles are not copied, backed up or deleted by SAMS:Disk.

If this sysparm is specified with a value of Y, SAMS:Disk will not create any profiles for data sets being backed up. SAMS:Disk will create a profile for data sets being archived.

If this sysparm is specified with a value of F, SAMS:Disk will not create any SAMS:Disk profiles, and in this case, it is the user's responsibility to ensure that a correct generic or discrete profile is available for any data set being restored prior to any user applications accessing that data set.

RACFDAEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACFDAEX user exit. Turn to the user exit description for *RACFDAEX* on page [256](#) for more information.

RACFDPEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACFDPEX user exit. Turn to the user exit description for *RACFDPEX* on page [257](#) for more information.

RACFDVL2dmsxxx

In order for PDS Compress processing to preserve data set profiles, SAMS:Disk temporarily adds a dummy volume to the RACF profile of the data set being compressed. This prevents the data set profile from being deleted when the data set is scratched and reallocated. The dummy volume is then automatically deleted by SAMS:Disk. This sysparm specifies the volume SAMS:Disk adds to the data set profile. The volume need not be a real DASD volume. Beware that if a real DASD volume is specified, it must not be a volume that will be processed by PDS Compress. The default value is DMSXXX.

RACFDVOLvvvvvv

This sysparm specifies the real online volume that is associated with all SAMS:Disk-saved RACF profiles. Specify this sysparm with a stable DASD volume name, such as, SYSRES, MSP001, CAT001, RESCUE or similar one- to six-character volume name.

It is a RACF requirement that discrete RACF profiles be associated with a real on-line DASD volume. No I/O or RESERVEs are done to this volume, its sole purpose is to satisfy the RACF requirement.

Under normal circumstances, this sysparm value should not be changed. However, if you have a compelling reason to change it, perhaps because the old volume is being disconnected, see *"Changing the Value of RACFDVOL"* on page 415 for related information.

RACFEAEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACFEAEX user exit. Turn to the user exit description for *RACFEAEX* on page 259 for more information.

RACFENTRY2

Specify this sysparm to select the SAMS:Disk RACF data set name encoding technique. Only two techniques are supported, 1 and 2. Encode technique 2 is the default.

Note: The default used to be 1. In Release 7.0 the default was changed to 2. For further information on encoding techniques, please review the topic *"Reconciling SAMS:Disk and RACF Records"* on page 395 in this manual.

RACFEPEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACFEPEX user exit. Turn to the user exit description for *RACFEPEX* on page 259 for more information.

[illegible]

See sysparm RACFMODL. It describes the conditions in which this sysparm should be used to provide the name of a model profile.

RACFMODLn

SAMS:Disk restore processing can fail if a discrete profile that was saved during archive is not available when it is being restored.

Specify this sysparm with a value of Y to have SAMS:Disk copy a model discrete profile instead of the unavailable saved profile. Using this model profile allows the restore to complete, after which you can correct its profile.

If you specify this sysparm with a Y, you must create the model profile to be used if it does not already exist. Then specify the data set name of the model profile as the value of sysparm RACFMDSN, and the volume of the profile in sysparm RACFMVOL. (A data set does not need to exist for the model profile.)

RACFMVOLvvvvvv

See sysparm RACFMODL. It describes the conditions in which this sysparm should be used to provide the volume of a model profile.

RACFNEWNn

Normally, SAMS:Disk does not allow RACF-protected data sets to be renamed with SAMS:Disk functions that support the NEWNAME command parameter. If users wish to allow SAMS:Disk to rename RACF-protected data sets, specify this sysparm with a value of Y. Its default value is N.

RACFPDSWn

Normally, SAMS:Disk does not explicitly protect the PDS compress work file. This creates a slight security exposure in the event that an abend or operator cancel leaves the work file on disk.

Specify this sysparm with a value of Y to instruct SAMS:Disk to protect the work file. SAMS:Disk will then set the RACF indicator bit in the format-1 DSCB of the work file to the same setting of the data set being compressed. If the data set has a discrete profile, SAMS:Disk will build an equivalent discrete profile for the work file. The work file will thus have protection equivalent to the data set being compressed.

If you wish to compress data sets covered by generic profiles or by RACF Global Access Table entries (and not covered by discrete profiles) and you also wish to set this sysparm to Y to protect the work file, you must create a generic profile or RACF Global Access Table entry to cover the work file. This is because SAMS:Disk does not copy generic profiles, and at least one profile must exist to cover the work file. This generic profile or table entry could be very restrictive — it need only allow access to the compress job and any potential reload jobs. This will ensure protection for all PDSs that may be compressed.

RACFPREDn

When restoring data sets protected by discrete profiles, specify a value of Y for this sysparm to have SAMS:Disk preserve already-existing discrete profiles. This option includes restoring to preallocated data sets, and restoring to non-preallocated data sets where discrete profiles already exist (perhaps in volume recovery situations when the RACF data set is still complete).

In the case of restoring to non-preallocated data sets where discrete profiles already exist, preserving the existing discrete profile is dependent on ADSP. If ADSP is active on the system and the RACF user has the ADSP attribute, SAMS:Disk will delete the existing discrete profile and create a new profile, using the SAMS:Disk-

saved profile as the model. If SAMS:Disk determines that ADSP is not active, it will preserve the existing discrete profile.

The default value of N, causes SAMS:Disk to delete any existing discrete profiles. If saved SAMS:Disk profiles are available, these replace the existing profiles. Carefully coordinate the specification of this sysparm with that of sysparm RACF-BKUP.

RACFPROC_y

When this sysparm is specified with the default value of Y, SAMS:Disk processes RACF-indicated data sets. We recommend this option if you have RACF installed at your installation. RACF users should also see sysparm RACFSUPP.

Specify a value of N to cause SAMS:Disk to bypass processing of RACF-indicated data sets. Normally, this value should not be used if you have RACF installed at your installation.

RACFSEQM_n

Normal sequential migration to tape processing will bypass RACF-protected data sets. If you want SAMS:Disk to attempt to process these data sets, specify a value of Y for this sysparm. The default value is N. Note: SAMS:Disk does not perform discrete profile checking or copy discrete profiles to the tape data sets when a value of Y is specified.

RACFSEQT_y

If sysparm RACFSEQM is specified with a value of Y and this sysparm is specified with a value of N, the output tape created by sequential migrate will not be RACF-indicated. The default value is Y, which causes tapes containing RACF-indicated data sets to be RACF- indicated.

RACFSMEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACFSMEX user exit. Turn to the user exit description for *RACFSMEX* on page [260](#) for more information.

RACFSUPP_n

When this sysparm is specified with the default value of N, the SAMS:Disk RACF support is OFF. SAMS:Disk does not check RACF authority or use SAMS:Disk RACF profile handling. SAMS:Disk tries to process RACF-protected data sets without interfacing to RACF. When this sysparm is specified with a value of Y, the SAMS:Disk RACF support is ON. SAMS:Disk checks RACF authority and uses SAMS:Disk RACF profile handling. Make sure that RACFPROC is specified with a value of Y.

RACFTESTn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

RACFUSIDxxxxxx

This sysparm is used to specify a 1- to 8-byte data set name prefix, which will be followed by a period and used as the first qualifier of the encoded data set name of SAMS:Disk-saved profiles. It can also be used to identify SAMS:Disk profiles and to allow them to be placed on a RACF data set apart from the standard RACF data set. This sysparm is required for SAMS:Disk discrete RACF profile support. The value of this sysparm should be chosen with care. Once specified, this prefix should not be changed.

It is a RACF restriction that this RACFUSID value represent a user ID or group ID. We recommend using a user ID, not a group ID, for the value for this sysparm. It should help you to identify SAMS:Disk profiles. We recommend using “DMSOS”. To avoid having RACF RACDEF processing update the PERMIT list, ensure that the user ID does not have the GRPACC attribute.

This RACF user ID will be able to restore any data set for which SAMS:Disk has saved a discrete profile. To prevent this exposure of unauthorized use of this RACF user ID, you can revoke it with the TSS command below. Revoking the RACF user ID in this manner will not prevent its use for SAMS:Disk discrete profile support.

```
ALTUSER racfusid REVOKE
```

RACFVCAVn

This sysparm indicates how the Move/Copy function is to create RACF discrete profiles for newly created data sets on target volumes. The default value is N, indicating that a RACF RACDEF will be done to create new profiles that point to the new data sets and volumes. A value of Y indicates that a RACF RACDEF with ADDVOL will be done to add the new volumes to the profile that already exists for the source data sets. ADDVOL processing will only be done in cases where the data set name for the source and target data sets are the same.

Note: ADDVOL processing will help to preserve RACF statistics maintained in RACF profiles since a RACDEF of a new profile loses statistics.

RACGDGVLvvvvvv

This sysparm is active only for sites with an active RACF support of SAMS:Disk.

GDG generation data sets can be RACF-protected by discrete profiles. Some releases of RACF also allow creating one model GDG profile with a name of the GDG base name that will cover each of the generations. This simplifies the security job, because only one profile is maintained instead of many. GDGs can also be protected by generic profiles.

Model GDG profiles are associated with a volume, just like discrete profiles. Generic profiles are not associated with a volume. At most installations, the volume associated with a model GDG profile is the same as the volume containing the catalog in which the GDG base is defined. By default, SAMS:Disk uses the catalog volume to determine the existence of model GDG profiles. SAMS:Disk does not copy model GDG profiles in archive, restore or any other functions.

Some installations have implemented non-standard model GDG profile handling code in their operating systems that causes all model GDG profiles in the shop to be associated with one single volume name. If your installation uses model GDG profiles and uses the non-standard support so that all model GDG profiles are associated with one volume, specify this sysparm with a value of that volume name. SAMS:Disk will use that volume in determining the existence of model GDG profiles.

RACHKAEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACHKAEX user exit. Turn to the user exit description for *RACHKAEX* on page [261](#) for more information.

RACHKPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RACHKPEX user exit. Turn to the user exit description for *RACHKPEX* on page [262](#) for more information.

RADEFAEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RADEFAEX user exit. Turn to the user exit description for *RADEFAEX* on page [263](#) for more information.

RADEFPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RADEFPEX user exit. Turn to the user exit description for *RADEFPEX* on page [264](#) for more information.

RADELAEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RADELAEX user exit. Turn to the user exit description for *RADELAEX* on page [265](#) for more information.

RADELPEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RADELPEX user exit. Turn to the user exit description for *RADELPEX* on page [266](#) for more information.

RADELSVCnnn

If SAMS:Disk saves discrete RACF profiles (see sysparm RACFBKUP), and users have been given the SAMS:Disk PFD function to delete index records for these data sets (see function *ADSINDEX* in the "*Defining PFD and DSCL User Options*" topic beginning on page 67 of the *Installation Guide*), users may have trouble deleting the DSNINDEX records for which SAMS:Disk has saved a discrete profile.

To keep the records synchronized, when SAMS:Disk is told to delete a DSNINDEX record, SAMS:Disk tries to delete the SAMS:Disk-saved profile, if any. Deleting a discrete profile requires APF-authorization (in fact, the system will abend non-APF-authorized users who attempt it). Most installations run their TSS and PFD sessions non- APF-authorized, and therefore cannot delete profiles unless APF-authorized by some means.

This sysparm should not be specified if you:

1. do not have discrete RACF profiles
2. specify sysparm RACFBKUP with a value of F
3. do not use the ADSINDEX option of SAMS:Disk PFD support,
4. run your PFD sessions APF-authorized

If you do not meet any of these criteria and your installation maintains an SVC for authorizing PFD sessions, specify this sysparm with a value representing your authorization SVC's 3-character number (SAMS:Disk will issue that SVC to get authorization).

If you do not meet any of these criteria and your installation does not maintain an SVC for authorizing PFD sessions, any SAMS:Disk attempt to delete a SAMS:Disk-saved profile will abend. In this case, remove option ADSINDEX from use by your users.

REBLIMITnnnnn

The value of this sysparm controls which technique SAMS:Disk will use to find a new blocksize when reblocking data sets. If the target blocksize specified is below this value, the algorithm is very straightforward; simply combine logical records into a block until one more would exceed the target blocksize.

If the target blocksize is above the value specified, SAMS:Disk uses a "best fit per track" algorithm, taking into account the key lengths and overhead bytes per block that may vary for each data set. The track size for the target device is divided by the target blocksize to give a target goal of xx blocks per track. SAMS:Disk then uses the LRECL, key length and overhead values to find the actual blocksize that will still yield the goal of xx blocks per track.

The default value of this parameter is 99999, such that the first algorithm is always used.

REBLKBITn

The DSEF system may determine what the blocksize of a data set should be during allocation. Data sets that may be reblocked by the system have the DS1REBLK bit on in the DS1SMSFG field of the format-1 DSCB. SAMS:Disk will honor this bit during allocation depending on sysparm REBLKBIT.

When this sysparm is specified with the default value of N, SAMS:Disk does not honor the reblock bit. The data set is not reblocked by the system.

When this sysparm is specified with a value of Y, SAMS:Disk honors the reblock bit during allocation. During a restore allocation, the reblock bit is honored only if the data set was archived with release 8.1 or above.

REBYPASSy

Specify a value of N for this sysparm if no messages are to be issued to explain why retain has bypassed a data set. The default value of Y causes these messages to be printed. This default value was chosen because the retain function is used for backup, and any data sets being bypassed should be known. Once these are determined to be all right, it may be desirable to suppress the messages.

RECATPSUr

When recataloging to the pseudo-volume, this sysparm's default value of R will cause SAMS:Disk to recatalog all data sets, whether or not they were originally cataloged. Specify a value of N to not recatalog data sets if they were not cataloged originally.

RECATVOLvvvvvv

Specify the volume serial to be used when the RECATALOG parameter is specified either on an ARCHIVE command or in the RETAIN volume specification table. The default volume serial is ARCIVE. If the value of the parameter is blanks (for example, "RECATVOL "), recatalog processing is not done. For related information, see sysparm PSUDEVTP on page 164 in this manual.

RECCHCATn

This sysparm is used for volume selection for non-VSAM data sets during the Recover function. Usage of the VOL= parameter will bypass this sysparm. When the default setting of 'N' is used, the volume recovery request will recover each data set to the specified volume, volume pool, or to the original volume. Specify a value of 'Y' to instruct SAMS:Disk to recover the data set using its cataloged volume.

RECPRIEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RECPRIEX user exit. Turn to the user exit description for *RECPRIEX* on page 267 for more information.

REDEVDEFn

This sysparm is used in conjunction with sysparm PSUDEVTP described on page 164 in this manual. When PSUDEVTP is specified with a value of Y, SAMS:Disk attempts to use the original volume's device type when recataloging a data set. This can cause a problem if the original volume is no longer online. (This condition applies primarily to the IXCATLG utility described in *"Auto-Restore Implementation Guidelines"*, beginning on page 59 of the *Installation Guide*).

If this sysparm is specified with a value of N, the default value, and the original volume is no longer online when SAMS:Disk attempts to recatalog a data set, SAMS:Disk will issue an error message. When this sysparm is specified with a value of Y, SAMS:Disk will use a default device type (F6425) if the original volume is not online.

RESAFTEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RESAFTEX user exit. Turn to the user exit description for *RESAFTEX* on page 268 for more information.

RESCHCATy

This sysparm is used for volume selection for non-VSAM data sets during the Restore function. Usage of the VOL= parameter will bypass this sysparm. When the default setting of 'Y' is used, the restore request will restore the data set to its cataloged volume(s). Specify a value of 'N' to instruct SAMS:Disk to skip the catalog check and to restore the data set to the specified volume, volume pool, or its original volume(s).

The value of this sysparm does not affect the auto-restore function since auto-restore uses different logic to determine the target volume during restore. For further information, review the topic *"The Auto-Restore Function"* beginning on page 42 of the *Installation Guide*.

RESCOPYTn

Specify a value of Y to instruct SAMS:Disk to use the duplicate copy tapes for all data set restores, including volume recovery. Default processing calls for the primary archive tape unless it has been disabled.

RESCRCATn

This sysparm controls scratch processing for both Restore and Recover functions for preallocated non-VSAM data sets. Invocation of this sysparm occurs when the SCRATCH parameter is specified.

Table 4-8. Possible Results when using RESCRCAT

Data Set Type	Results
Uncataloged data sets	When the default setting of 'N' is used, SAMS:Disk will scratch the uncataloged preallocated data set if it resides on the same volume as the one to which allocation is being directed
Cataloged data sets	When the default setting of 'N' is used, SAMS:Disk will delete the cataloged data set if the cataloged volume is the same as the one to which allocation is being directed
Cataloged data sets only	When 'Y' is specified, SAMS:Disk will delete the cataloged data set prior to volume selection

See the description of sysparm DSNDUPVL on page 145 for related information.

RESIXDLBn

This sysparm is used only in conjunction with RESIXRPD, and controls processing of the backup DSNINDEX record following a successful restore or recover. Backup DSNINDEX records do not have the scratch flag turned on (the data set was not scratched at archive time). The default value of N indicates no change is to be made to the backup DSNINDEX record. Specify a value of Y to cause the backup DSNINDEX record to be processed in the same manner as the archive DSNINDEX record after a successful restore or recover (as controlled by RESIXRPD).

For discussion of this sysparm, see “Using Archive Functions to Manage Active Data” on page 67 in this manual.

RESIXRPDnn

This sysparm helps control the management of active data sets through the archive function; that is, placing data sets that are used only occasionally in the SAMS:Disk archives in order to conserve disk space. (These “active” data sets are usually archived to disk in a compressed format, and should be thought of as temporarily compressed rather than archived. See “Using Archive Functions to Manage Active Data” on page 67 for a more detailed explanation of this procedure.)

The default value of 99 for this sysparm indicates that restore processing should not change the expiration date of a data set in the SAMS:Disk archives.

Specify a value of 00 to delete the data set from the SAMS:Disk archives after an archived data set is restored or recovered successfully under its original name.

Specify a value of 01 to 98 to have SAMS:Disk assign a new expiration date to the data set in the SAMS:Disk archives after it is restored or recovered successfully under its original name. (See also sysparm *IXMCLEAN* on page 155 in this manual). The new expiration date for the archived data set is the date it is restored plus the number of days specified by this sysparm. (You might choose this last option to allow the data set to be backed up before the earlier archive rolls out of the SAMS:Disk archives.)

To prevent a potential loss of archived data, if any of the NOLOAD, NEWNAME, MEMBER or AIXNAME parameters are specified on a Restore command, this function is not active and no change is made to the expiration date of the DSNINDEX record.

This sysparm may be used in conjunction with RESIXDLB to control the processing of DSNINDEX records associated with backup copies; that is, should the DSNINDEX record be deleted or reset after a restore from a backup copy of the data set.

RESPRIEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RESPRIEX user exit. Turn to the user exit description for *RESPRIEX* on page 269 for more information.

RESRETPDnnnnn

This sysparm controls the number of days that a restored data set is to be exempt from archive disposition, and can be specified with a 5-digit numerical value. The default “grace period” is 00030, but is activated only if sysparm RETEXCLD is specified with a value of Y.

RESTCTLGy

This sysparm may be specified with a value of N to suppress catalog referencing during preparation of the Restored Data Set report. The default value of Y will cause SAMS:Disk to do catalog locates for every data set restored.

Note: If a large number of data sets are being restored (for example, volume recovery), checking the catalog incurs significant overhead. If this information is not needed on the report, change the value to N to cause significant performance improvements in report processing.

RETAFTExmmmmmmmmmm

Specify this sysparm to provide the 8-character name of the user-supplied exit module to receive control after data set disposition determination by SAMS:Disk. Turn to the user exit description for *RETAFTEx* on page 270 for more information.

RETDSNCDccasc

This sysparm controls retention control at the data set level, based on the creation date. Specify for “cc” the year code that must be matched to the data set’s expiration

year (code). Specify for “asc” the desired archive, scratch and catalog actions to be taken if the data set’s create date plus (ddd) days (from the expiration date field) has expired.

RETDSNCLccas

This sysparm controls retention control at the data set level, based on catalog status. Specify for “cc” the year code that must be matched to the data set’s expiration year (code). Specify for “as” the desired archive and scratch actions to be taken if the data set is not cataloged.

RETDSNDFccddd

This sysparm is used only within the data set level control processing option of retention control. Specify for “ccddd” the default expiration date that is to be given to all data sets that have a zero expiration date field. This control code and retention period are then used to control the analysis and processing as called for by the three data set level parameters.

RETDSNLUccasc

This sysparm controls retention control at the data set level, based on last used date. Specify for “cc” the year code that must be matched to the data set’s expiration year (code). Specify for “asc” the desired archive, scratch and catalog actions to be taken if the data set’s last used date plus (ddd) days (from the expiration date field) has expired.

RETEXCLDn

Specifying this sysparm with a value of Y causes a “grace period” record to be placed in the RETEXCLD file each time a data set is restored. The presence of such a record will prevent retention control from automatically disposing of a restored data set until the grace period of 30 days has elapsed. The default value of this sysparm is N. The default grace period value is controlled by sysparm RESRETPD.

The grace period record contains the data set name and the volume to which it was restored. A data set with the same name but on a different volume is regarded as a different data set, and will therefore not be exempt from retention control processing. For this reason, moving a restored data set to a new volume effectively terminates its grace period.

RETEXCMDn

This sysparm only has effect when sysparm RETEXCLD is specified with a value of Y. The default value of N indicates that users cannot override the restore default grace period (specified by sysparm RESRETPD) with the RETPD and EXPDT restore command parameters. If a value of Y is specified for this sysparm, the RETPD and EXPDT command parameters will override the default grace period.

The RETPD and EXPDT command parameters are not documented for the RESTORE and DRESTORE commands. They may be specified if this sysparm is specified with a value of Y.

RETPD is the number of days that the data set remains excluded from archive processing from the date of the restore. Specify 1 to 5 numeric digits (that is, RETPD=7). EXPDT is a Julian date that may be specified instead of RETPD. It has the same effect but is expressed in the form of a date (that is, EXPDT=1983.365).

RETGDGEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RETGDGEX user exit. Turn to the user exit description for *RETGDGEX* on page [271](#) for more information.

RETGDGNMxxxxxxxxx

When the G selection criteria is used in the retain function, different limits for different generation base names may be specified in a GDG DASD generations member of the parmlib data set. To activate the list, specify this sysparm with the name of the member.

RETMULTIn

Specify a value of Y for this sysparm if multivolume PS and DA data sets are to be processed by requests processed by the RETAIN proc. The default value is N, not to process multivolume data sets.

See sysparm *ARCMULTI* on page [127](#) for controls on archives done by the ARCHIVE proc. There are no controls on archives done by the DMS proc.

RETNODSPyy

Specifying a year value in this sysparm will preclude retention control disposition of any type for any data set having the same year code with zero (ddd) in its expiration date field.

RETNOMIGyy

Specifying a year value in this sysparm will preclude sequential migration to tape of any data set having the same year code with zero (ddd) in its expiration date field.

RETPREFTc

The default value of C will cause SAMS:Disk to use the creation date for the data set from the VTOC entry for retention date calculation. Specify a value of U to have the retain volume specification type R selection criteria use the last use date from the VTOC entry for retention date calculation.

RETPRIEXmmmmmmmmmm

Specify for “mmmmmmmmmm” the name of a user-supplied exit module to receive control prior to inspection of the DSCB during retention control processing. Turn to the user exit description for *RETPRIEX* on page [272](#) for more information.

RETRETPDnnnnn

When data sets are selected to be archived during archive or backup processing, the 5-digit value of this sysparm is added to the current date to designate the archive retention period. This value is used unless an overriding parameter has been specified on the ARCHIVE or BACKUP command. The default value of this sysparm is 00030, to keep each data set in the archives for 30 days. See also sysparm *EXPDTUSE* on page 149 in this manual.

RLSEAFEXmmmmmmmm

Specify the 8-character name of the user-supplied exit module to receive control after a data set has been selected for idle space release processing. Turn to the user exit description for *RLSEAFEX* on page 273 for more information.

RLSEDIAGn

Specify a value of Y to cause a message to be printed stating why a data set was bypassed for Idle Space Release. The default value of N suppresses such messages.

RLSEMAXSnnnnn

This sysparm is used to specify the maximum number of tracks allowed when resetting the secondary space allocation for any given data set. The default value is 99999 tracks. Since the resetting of secondary allocation is based on a percentage of the new primary allocation, this allows the user to specify a ceiling secondary allocation value on data sets with a large primary allocation.

Note: This value is in tracks. Should the data set be allocated in cylinders, this value will be rounded up to the next whole cylinder, thus actually allocating more space than the maximum amount specified by this sysparm.

Example: On a F493, there are 30 tracks per cylinder. The user specifies a value of 00045 for this sysparm. If the data set is allocated in cylinders and the secondary allocation is computed to be equivalent to 35 tracks, this amount will be converted to cylinders and rounded up to the nearest whole cylinder, which would be two cylinders. On a F493, this is equivalent to 60 tracks, which is actually over the maximum amount specified by this sysparm.

RLSEMINSnnnnn

This sysparm specifies a minimum number of tracks allowed when resetting the secondary space allocation for any given data set. The default value is 00000 tracks. Since the resetting of secondary allocation is based on a percentage of the new primary allocation, this allows the user to specify a default secondary allocation value on data sets with a small primary allocation.

Note: This value is in tracks.

RLSESECRa

The default value, A, will cause SAMS:Disk to reset the secondary space allocation on all data sets (which meet all other RELEASE criteria) whenever specified via the

PCTSECONDARY parameter for DSCL or through the PSSEC= or POSEC= parameters for batch. Specify a value of Z to reset only those secondary allocations with current value of zero, or a value of N to reset only those secondary allocations that currently have non-zero values.

RLSESYS1n

Specify a value of Y to allow idle space release processing of SYS1 data sets. The default value of N prevents idle space release processing of SYS1 data sets.

RLSEUNMVy

Specify a value of N to prevent idle space release processing of unmovable data sets. The default value of Y will permit idle space release processing of unmoveable data sets.

RPTCLSELxx...x

This sysparm may be used to specify the default group codes to be listed on the VSAM Cluster Detail report. Specify for the “xx.xx” string the desired groups, and the order in which they are to appear on the report. When this sysparm is not specified, a default value of U is assumed, causing use statistics to be printed.

For additional information, see the *CLDOPT* parameter description on page 106 in the *User's Guide*.

RPTDSBEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RPTDSBEX user exit. Turn to the user exit description for *RPTDSBEX* on page 274 for more information.

RPTDSNEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RPTDSNEX user exit. Turn to the user exit description for *RPTDSNEX* on page 274 for more information.

RPTDSUEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RPTDSUEX user exit. Turn to the user exit description for *RPTDSUEX* on page 276 for more information.

RPTLINES

This sysparm, which defaults to 058, controls the number of lines per page for the following reports:

ALLOCS	LISTD	PDS COMPRESS
ATTRBS	LISTV	
DISTR	MAPPER	Utilization by DATE
DSINDX	MEMBER	and TIME
DSUTIL	MVDICT	
FREESP	PDSTAT	
ISAMDS	SVDICT	
LAYOUT		

Figure 4-2. List of Reports Controlled by RPTLINES

All other reports may be controlled via the following zap.

```
NAME ADSUT190 LINEMAX
VER 0000 058C      58 lines per page
REP 0000 0nnC      nn lines per page
```

RPTMVDEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the RPTMVDEX user exit. Turn to the user exit description for *RPTMVDEX* on page [276](#) for more information.

RPTNMTSZxxx

Specify this sysparm with a numeric value to represent the number of reports included in the parmlib member REPORTNM. The default value is 100. If the value of this sysparm is exceeded, an error message will be issued stating that this value should be increased.

RSCHGBITn

Specify a value of Y to cause SAMS:Disk to set the change bit on each restored data set. Setting the change bit on each restored data sets may increase the number of data sets backed up in implicit archive runs, but may simplify the recovery of packs. The default value of N causes SAMS:Disk to leave the change bit set the same as before the data set was archived or backed up. See also sysparm *VCCHGBIT* on page [197](#) in this manual.

RSUPPRESn

Specify a value of Y to suppress issuance of the RESERVE macro on the volume where the files data set resides during index maintenance functions. The integrity of the files data set cannot be guaranteed if this parameter is specified with a Y. In a JES environment, the user is responsible for ensuring that no other updates are occurring through a cross system enqueue package. The default value is N.

SCRASYS1n

With the default value of N in effect for this sysparm, SAMS:Disk will not allow users to scratch any data sets prefixed by “SYS1.” through its facilities. This is to prevent users from inadvertently scratching data sets critical to the operating system. To override this feature and allow SAMS:Disk to scratch data sets prefixed by “SYS1.”, specify a value of Y.

SCRTCHEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the SCRTCHEX user exit. Turn to the user exit description for *SCRTCHEX* on page 277 for more information.

SECURAEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the SECURAEX user exit. Turn to the user exit description for *SECURAEX* on page 278 for more information.

SECURCMDn

To activate the security validation of SAMS:Disk commands, specify Y as the value of this sysparm. The default value of N indicates that no security checking is to be done on SAMS:Disk commands. This feature is active only when the RACF support is active.

SECURDRSaaaaaaa

This sysparm allows the user to specify the level of authority SAMS:Disk will check for in a deferred RESTORE request. Authority levels of OUTPUT, SCRATCH, and ALLOCATE may be specified. The default is OUTPUT. If SECURDRS is set to OUTPUT, and the SCRATCH parameter has been specified on the command, SAMS:Disk will check for both OUTPUT and SCRATCH authority to the data set. OUTPUT, ALLOCATE and SCRATCH are internal SAMS:Disk authorization levels. They correspond to authorization levels of the security packages as described below.

Table 4-9. SECURDRS Authorization Levels

SAMS:Disk	RACF
OUTPUT	UPDATE
SCRATCH	ALTER
ALLOCATE	ALTER

SECURFILn

To activate the security validation of access to subfiles of the files data set, specify a value of Y for this sysparm. The default value of N indicates that no security checking is to be done. This feature is active only when the RACF support is active.

SECURLIMnnnn

This sysparm may be used to limit the number of security violations permitted by the SAMS:Disk Security Interface. The default value of 0000 will allow SAMS:Disk to continue processing, regardless of the number of violations detected by the Security Interface.

Specify a non-zero 4-digit value for this sysparm to have SAMS:Disk issue message 2582 and abend user 913 if the SAMS:Disk Security Interface detects more violations than the value of this sysparm. Security violations generated by system operations, such as OPEN or SCRATCH, will not be counted by this processing, and will not cause the message 2582 and abend to occur.

SECURLOGy

In each SAMS:Disk Security Interface, SAMS:Disk queries your security package for authority to process a resource. SAMS:Disk then examines the return code from your security package. If authorized, SAMS:Disk continues with the processing of that resource. If not authorized, SAMS:Disk prints an appropriate message and bypasses that resource. Either way, SAMS:Disk continues processing the next resource as required by the SAMS:Disk commands.

In addition to the SAMS:Disk authorization failure message, by default each security package issues its own authorization failure message to the system log, the security console, and to the TSS user or job log. It may also create SMF records for a report writer indicating access violations. If you wish to inhibit RACF from issuing its own authorization failure messages or creating SMF records during queries from their respective SAMS:Disk Security Interfaces running as APF-authorized tasks, specify this sysparm with a value of N.

To prevent system abends, if SAMS:Disk is running as a non- APF-authorized task, such as under most TSS and PFD sessions, SAMS:Disk will ignore the setting of this sysparm and allow system logging.

Regardless of the value of this sysparm, each security package will continue to issue its own authorization failure messages for authorization failures in SAMS:Disk processing that are not part of the SAMS:Disk Security Interfaces, or for authorization failures in non- APF-authorized tasks, such as under most TSS and PFD sessions.

SECURPEXmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the SECURPEX user exit. Turn to the user exit description for *SECURPEX* on page [279](#) for more information.

SECURRPTn

To activate security validation during SAMS:Disk report generation (including the reports that list the ARCHVOLS records), specify a value of Y for this parameter. The default value of N indicates that no security checking is to be done. This feature is active only when the RACF support is active.

SECURVOLy

If this sysparm is specified with the default value of Y, and you have sysparm RACFSUPP specified with a value of Y, and you have the RACF DASDVOL class active, SAMS:Disk will query your security package for volume-level (SCRATCH operations only) authority during data set-oriented functions and volume-oriented functions.

If you specify this sysparm with a value of N, or you do not have sysparm RACFSUPP specified with a value of Y, or you have the RACF DASDVOL class inactive, SAMS:Disk will not query your security package for volume-level (SCRATCH operations only) authority during data set-oriented functions and volume-oriented functions.

SECUSUPPn

The default value for this sysparm is N which, in conjunction with sysparm PASSWORD, causes SAMS:Disk to bypass security processing for all password-indicated non-VSAM user data sets when they are encountered. This avoids prompting of the operator for the passwords to password-indicated non-VSAM user data sets.

If this sysparm is specified with a value of Y, SAMS:Disk will not bypass security processing for password-indicated non-VSAM user data sets. This will allow the security package SECURE to check authorization in OPEN of user data sets if you have SECURE installed. It will also allow the operator to be prompted for passwords in OPEN of user data sets if you do not have SECURE installed.

If you have the security package SECURE installed, you should specify this sysparm with a value of Y. You should also consider specifying sysparm PASSNEWN. Please refer to the topic *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide*.

SELECTREy

The intention of this sysparm is to help prevent runaway jobs from occurring. With the default of Y specified, DSCL requires that at least one SELECT statement be included on every SCAN command. (This sysparm has no effect on the FMS PROC). Regardless of its value, no SELECT statements are required for the FIND command. However, if you specify a value of N and fail to provide a SELECT statement on the SCAN command, SAMS:Disk will attempt to process ALL data sets found by the SCAN. This can allow runaway jobs to occur! For example, if a user unintentionally runs:

```
SCAN REALVOLS  
DELETE
```

Note: For other related sysparms, refer to the descriptions of *DSNSELRE* (page 145 in this manual) and *VOLSELRE* (page 198 in this manual).

SMFCDATDnnn

The displacement of the close date in the SMF record is indicated as the 3-digit value for this sysparm. The default value is 006.

SMFCTIMDnnn

The displacement of the create time in the SMF record is indicated as the 3-digit value for this entry. The default value is 002.

SMFDFLAGnnn

The displacement of the flag byte in the SMF record that indicates a DASD data set is specified by the 3-digit value of this sysparm. When 000 is specified, all records will be treated as DASD. The default value is 038.

SMFDSNMDnnn

The displacement of the data set name in the SMF record is indicated as the 3-digit value for this entry. The default value is 064.

SMFJDATDnnn

The displacement of reader/interpreter job date in the SMF record is indicated as the 3-digit value for this entry. The default value is 026.

SMFJTIMDnnn

The displacement of reader/interpreter job time in the SMF record is indicated as the 3-digit value for this entry. The default value is 022.

SMFTYP14nnn

The value specified for this sysparm identifies the TYPE 14 data set activity record. The value must be specified from the range of 001 through 255. The default value is 014.

SMFTYP15nnn

The value specified for this sysparm identifies the TYPE 15 data set activity record. The value must be specified from the range of 001 through 255. The default value is 015.

SMFVLSRDnnn

The displacement of the volume serial in the SMF record is indicated as the value of this parameter. The default value is 182.

SORTPADdy

If the installation sort routine allows a sort key length greater than the actual record length for variable-length records, you may specify a value of N for this sysparm to prevent SAMS:Disk from padding all report records to a length equal to the key. The default value of Y permits the padding of the report records.

SPACECEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the SPACECEX user exit. Turn to the user exit description for *SPACECEX* on page 283 for more information.

SPACEPRInnn

When archiving to disk, you must specify a primary space allocation value for the archive data set. The default value of 999 causes SAMS:Disk to set the primary quantity equal to 1/16 of an assumed backup tape. You may specify a value from 1 to 998, which represents the number of megabytes to allocate.

SPACESECnnn

When archiving to disk, you must specify a secondary space allocation value for the archive data set. The default value of 999 causes SAMS:Disk to set the secondary quantity equal to 1/16 of an assumed backup tape. You may specify a value from 0 to 998, which represents the number of megabytes to allocate.

SPFAUNITnnnnnnnnn

Specify this sysparm to change the unit type allocated for the work data sets used in PFD online reporting. The default unit name is SYSALLDA.

SPFDSNAMnn....nn

This sysparm is used to specify a system-wide data set to be used to store PFD report definitions. This is the same data set name that is specified when you create the report definition data set, either using the PFD online facility or the batch utility provided (review the topic *"Overview of PFD Online Reporting Feature"* beginning on page 455 in the *User's Guide* for more information on creating this data set). The default data set name can be overridden both on the primary online report panel and in any batch job that uses the PFD reporting feature. It is used only with the PFD support.

SPFENQQNnnnnnnnnn

When saving JCL in the SAMS:Disk PFD support, SAMS:Disk ENQs with a DISP=OLD on the data set in which the JCL is to be saved. The QNAME used is SPFDSN. If you desire to change the QNAME, specify this sysparm with another name, using a maximum of 8 characters.

SPFENQTYr

When saving JCL in the SAMS:Disk PFD support, SAMS:Disk ENQs with a DISP=OLD on the data set in which the JCL is to be saved. If you do not wish SAMS:Disk to do the ENQ, specify this sysparm with a value of N.

Note: Specifying this sysparm with a value of N is an integrity exposure in that two or more persons could possibly update the same data set, thus destroying data.

SPFENQWTnn

When processing partitioned data sets, Fujitsu's PFD editor and linkage-editor routines use enqueue conventions beyond the standard data set name enqueue under QNAME of SYSDSN. Therefore, while processing partitioned data sets, SAMS:Disk also issues an exclusive enqueue on the data set name with QNAME of SPFDSN (this QNAME value can be changed with sysparm SPFENQQN). If the record format of the data set is also undefined, an exclusive enqueue on the data set name and QNAME of SYSIEWLP is done as well. These enqueues are done to prevent a data loss that can occur if another task is processing the data set at the same time SAMS:Disk is processing it.

Specify this sysparm with a one- or two-digit value to indicate the number of seconds the enqueue routine will wait for the exclusive enqueues. The default value is four seconds. If the enqueues are not obtained within the specified time limit, the data set will not be processed. If a value of zero is specified, no enqueue processing for the QNAMEs of SPFDSN or SYSIEWLP will be done.

This processing is not affected by the SAMS:Disk user exit USERSENQ.

Note: Specifying this sysparm with a value of zero is an integrity exposure, in that two or more tasks could possibly update the same data set, thus destroying data.

SPFHDXGEXmmmmmmmmmm

This sysparm, used only with the PFD support, specifies the module name to be invoked for the SPFHDXGEX user exit. Turn to the user exit description for *SPFHDXGEX* on page [284](#) for more information.

SPFHILVNnnnnnnnnnn

Specify this sysparm to use a high-level node (maximum of 8 characters) in the work data sets used in PFD online reporting. The default is spaces.

SPFORDEXmmmmmmmmmm

This sysparm, used only with the PFD support, specifies the module name to be invoked for the SPFORDEX user exit. Turn to the user exit description for *SPFORDEX* on page [285](#) for more information.

SPFORSEXmmmmmmmmmm

This sysparm, used only with the PFD support, specifies the module name to be invoked for the SPFORSEX user exit. Turn to the user exit description for *SPFORSEX* on page [285](#) for more information.

SPFORSPCnnnn

Specify this sysparm to change the number of blocks allocated for the REPORTS ddname in PFD online reporting. The default is 150 blocks.

SPFORSPEnnnn

Specify this sysparm to change the number of blocks allocated for the SPFRPTS ddname in PFD online reporting. The default is 100 blocks.

SPFRPTDTy

This sysparm can be specified to suppress printing of detail report lines for the PFD online reporting facility. The default value of Y causes all detail lines to be printed. If you set the value to N, SAMS:Disk will print only subtotal and grand total lines. This feature is useful in place of the volume summary reports if you want to limit the totals to a subset of the data sets on the volume (with SELECT or EXCLUDE commands). If you want to override this parameter in the JCL with the SYSPARMS dd statement, you must do it in the print step (that is, PRINT.SYSPARMS DD *). It is used only with the PFD support.

SPFSYSINnnnnnnnnn

This sysparm may be specified to provide the ddname to use in place of SYSIN in the PFD support only. The default is SPFSYSIN. To change the ddname, specify this sysparm with another name (maximum of 8 characters).

SPFUSRIDn

This sysparm applies only to the PFD support. It may be specified with a value of Y to restrict PFD users to archive and restore only those data sets that begin with their user ID. You may exempt selected users from this restriction by placing their user IDs in the TSOUSERI member of the parmlib data set. The default value for this sysparm is N, which allows all users access to all data sets.

SVCNODMSnnn

Specify for “nnn” the decimal number of the SAMS:Disk SVC as installed. The default value is 244.

SYSOUTEXmmmmmmmmmm

Unlike other SAMS:Disk user exits, the SYSOUT exit is not activated by specifying a sysparm. The module name must be superzapped into the SAMS:Disk output writer module. This entry in the sysparm section is maintained solely as a reference aid. Turn to the user exit description for *SYSOUTEX* on page [286](#) for more information.

SYSPARMon

SAMS:Disk provides the capability for users to override default sysparm values and entries in the SYSPARMS member of the parmlib data set at execution time via the //SYSPARMS dd statement. This provides the flexibility of tailoring parameter values for various executions without having to update the parmlib data set. Since sysparms can affect the security and integrity of user data sets, this feature is controlled through this sysparm.

The default value of N prevents all users from overriding sysparms. To allow all users to override some or all sysparms, or to allow only authorized users this ability,

specify this sysparm with a value of Y. (See “Resources Protected by the SAMS:Disk Base System” on page 371 in this manual.)

For a further discussion on overriding sysparms, see “Overriding Sysparms In-stream” on page 104 in this manual.

SYSPARMSn

This sysparm controls the display of system parameters on the message dd statement during SAMS:Disk executions. It gives the user the option of seeing what sysparms were active during the SAMS:Disk run.

The sysparm contains a 1-byte value. The default value of N indicates no display is to be generated. Other valid values are:

Table 4-10. Possible Values Available for SYSPARMS

Value	Meaning
O	print only those sysparms overridden by SYSPARMS DD
S	print only those sysparms supplied by user
D	print only the default sysparms
B	print all sysparms in effect

TAOPPOEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the TAOPPOEX user exit. Turn to the user exit description for *TAOPPOEX* on page 287 for more information.

TAOPPREXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the TAOPPREX user exit. Turn to the user exit description for *TAOPPREX* on page 288 for more information.

TAPCTLEXmmmmmmmmmm

Specify this sysparm to provide the module name, other than ADSTH016, to be invoked for the TAPCTLEX user exit if more than one tape management system is used. For related information, turn to the sysparm description of *TMSCTLEX* on page 191 and the user exit description for *TAPCTLEX* on page 289 in this manual.

TAPEDLEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the TAPEDLEX user exit. Turn to the user exit description for *TAPEDLEX* on page [289](#) for more information.

TAPEFEETnnnn

The 4-digit value specified for this sysparm is the number of feet to be used for writing data sets on each reel of tape. In the event that a reel that has less than the specified number of feet is mounted, it is possible that a data set will overflow onto a second reel. However, the following data set will be archived to the beginning of another reel, precluding the possibility that more than one reel of tape would have to be passed to accommodate a subsequent restore request. The default value for this sysparm is 2300.

TAPEFSCRn

Specifying a value of Y allows SAMS:Disk to bypass expiration date and security checking when writing to a tape — in other words to “force” its use as a valid scratch tape. The standard operator prompt for permission to write on a date-protected tape will not be issued. Similarly, the operating system will not abend the job when a security-protected tape is mounted for output. This can be particularly useful when SAMS:Disk tape pools are used. (Tapes are recycled, and by definition, SAMS:Disk knows that it is okay to write on them again.) The default value is N, however, since using this facility with general scratch tapes represents the loss of a safety check, not to mention the data, if wrong tapes are mounted.

Note: If your tapes are managed by CA1, do not specify a value of Y, or all CA1 processing will be bypassed.

TAPEPOOLn

Specify a value of Y to cause SAMS:Disk to take output tapes for archiving out of the TAPEPOOL table in the DMSPOOLS subfile. The default value of N will call for the operator to mount any available scratch tape.

TAPEPROTn

When this sysparm is specified with the default value of N, SAMS:Disk does not password-protect archive data sets on tape and allows normal security checking when opening them.

When this sysparm is specified with a value of Y, SAMS:Disk sets the password-indicator on the tape label, but does not generate any passwords. Also, SAMS:Disk bypasses all security checking on the archive output tape volumes.

If this sysparm is specified with a value of Y, sysparm TAPEFSCR must also be specified with a value of Y to allow SAMS:Disk to reuse tapes. (At last report, this was incompatible with the CA1 tape management system from Computer Associates.) If a SAMS:Disk archive tape was created with sysparm TAPEPROT specified with a value of Y, you must relabel the tape when it is released from the

SAMS:Disk archives to turn off the password-indication of the tape label. This applies to SAMS:Disk as well as other applications.

For password-protection of archive data sets on disk, review the sysparm description for *DISKPROT* on page 141 in this manual.

TAPEPRTRy

Specify a value of N for this sysparm to cause SAMS:Disk to use normal security checking on reading archive tapes. SAMS:Disk will prompt the operator for the password for archive tapes created with sysparm TAPEPROT specified with a Y.

The default value of Y will cause SAMS:Disk to bypass security checking when reading archive tapes. SAMS:Disk will read archive tapes created with the sysparm TAPEPROT specified with a Y without operator intervention.

TAPEVVERY

The default value of Y causes SAMS:Disk to use the Fujitsu DSEF volume verification exit to verify that each tape mounted for output is indeed a scratch tape, (not in the SAMS:Disk archives already) and, if it is not, to dynamically request another tape. All of this can be done without damaging the mounted tape, and processing can continue.

If vendors of other products used in your installation state that they do not support this exit, this sysparm should be specified with a value of N to deactivate its use. When a “wrong tape” is mounted under these conditions, SAMS:Disk will issue appropriate messages and terminate processing in an effort to preserve the data already on the tape.

TAPMFEETnnnn

This sysparm specifies the number of feet to be used for writing data sets on each tape reel in the sequential migration to tape function. The default value is 2300. If a reel that has fewer than the specified number of feet is mounted, a data set could overflow onto a second reel.

TERMCLERy

The default action in TSS is to clear the terminal screen before displaying the output. Specifying a value of N will cause SAMS:Disk to not clear the terminal screen before displaying the output.

TESTMSGSn

This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Technical Support.

TMSCTLEXmmmmmmmm

Specify this sysparm to provide a module name for the TMSCTLEX user exit. Turn to the user exit description for *TMSCTLEX* on page 290 for more information.

Note: In order to activate the RMM support, this sysparm must be specified with a value of ADSTH016. See sysparm *TAPCTLEX* on page 189 if more than one tape management system is used.

TPMGMTExmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the TPMGMTEx user exit. Turn to the user exit description for *TPMGMTEx* on page 294 for more information.

TPOOLCHGn

This sysparm applies only to the PFD support. The online facility to list the tapes in a tape pool does not normally allow the “scratch” or “in-use” status of a tape to be changed. The entry may only be deleted (by placing a “D” beside it) and then added again as a scratch tape if desired. Specifying a value of Y for this sysparm permits you to enter an “A” beside the tape entry to change it to an available (scratch) status, or an “I” beside the entry to mark it as in-use. (Use this feature with extreme caution! Careless use can wreak havoc upon the tape pools in very short order.)

TPOOLSEQs

The default value of S for this sysparm keeps scratch tapes in the tape pool in sequence by when they were scratched. Specify a value of V to keep them in volume serial sequence. If you use the EDM interface, you should not set this sysparm to a value of V because SAMS:Disk may attempt to use the tape again before it has been scratched by the nightly TMSCLEAN run.

TRKOVCHGn

Specify a value of Y for this sysparm to cause SAMS:Disk to allow the track overflow attribute of a data set to be dropped when the target device does not support that feature (for example, F6425 devices) but the blocksize will fit on the target track without the attribute. This sysparm applies only to the MOVE/COPY DSCL commands.

The default value of N will cause SAMS:Disk to bypass these data sets with a diagnostic message such that the user may decide if these data sets are to be copied.

TSODACATu

The default value of U for this sysparm causes SAMS:Disk to uncatalog a data set when performing an PFD or TSS deferred archive. This action can be changed by specifying either a value of N for no catalog change, or a value of R to recatalog to the SAMS:Disk pseudo-volume.

TSODARESy

The default value of Y for this sysparm causes SAMS:Disk to reset (turn off) the change bit when doing an PFD or TSS deferred archive without scratching the data set. This action can be changed to not reset the change bit by specifying a value of N.

TSODASCRs

The default value of S causes SAMS:Disk to scratch a data set when doing an PFD or TSS deferred archive. This action can be changed to “no scratch” by specifying a value of N.

TSOTULMTnn

Specify for “nn” the maximum number of tape units that may allocated concurrently for TSS dynamic (immediate) restores. The default value is 01. When this number is reached, any subsequent TSS dynamic restore request will be denied with a message that the maximum numbered of tape units are already allocated.

TSOTUNITnnnnnnnn

Specify for “nnnnnnnn” the 1- to 8-character unit name for the class of tape devices to be used for dynamic allocation of SAMS:Disk archive tapes under MSP. The default unit name is TAPE.

TSOUSRIDn

This sysparm applies to the TSS support. It may be specified with a value of Y to restrict users queueing archive and restore requests to only those data sets that begin with their user ID. You may exempt selected users from this restriction by placing their user IDs in the TSOUSERI member of the parmlib data set. The default value for this sysparm is N, which allows all users access to all data sets.

TSOVOLOFn

The default value of N for this sysparm causes the TSS command processors to reject requests issued against volumes that cannot be found online. Specify a value of Y to allow requests to be made against data sets that cannot be found online (they are assumed to be contained on offline volumes).

TTRULMOVn

The default value of N for this sysparm prohibits SAMS:Disk from moving a data set with TTR addressing to a different device type. Specifying a value of Y allows SAMS:Disk to move a data set with TTR addressing to a different device type.

Note: Because of the unique nature of these types of data sets, it is imperative that the user verify the integrity of the data set after the move.

UCBALLOCxx

This sysparm controls the maximum number of volumes that a SAMS:Disk job may have dynamically allocated at one time. This corresponds to the number of UCBs in an MSP environment. The default value is 21. Change this limit by specifying a value of from 1 to 64 for this sysparm. We recommend specifying this sysparm to a value no lower than 3 (higher if you have specified sysparm RETMULTI).

UCC1BYP*Sn***

This sysparm is valid only when TMSCTLEX is specified with a value of ADSTH025. If you want to have SAMS:Disk keep running even if the CA-1 system is down, specify a value of Y. The default value of N will cause SAMS:Disk to abend with an appropriate message.

This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025".

UCC1EXPR*y*

This sysparm is valid only when TMSCTLEX is specified with a value of ADSTH025. The default value of Y will cause SAMS:Disk to expire tapes in the CA-1's TMC by setting the expiration date of the tape to the current date. Specify a value of N if sysparm UCC1SCRT is specified with a value of Y.

This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025".

UCC1EXTN*nnnn*

The default value of 0000 for this sysparm will cause SAMS:Disk (in CA-1 support) to expire a tape by setting the expiration date of the tape to the current date. If you want to allow a grace period before the tape is expired, specify a 1- to 4-digit number indicating the number of days to be added to the current date to determine the expiration of the tape in CA-1's TMC.

This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025".

UCC1SCRT*n*

The default value of N for this sysparm will cause SAMS:Disk (in CA-1 support) to expire a tape by setting the expiration date of the tape to the current date. If you want to have CA-1 recognize the tape immediately as a scratch tape, specify a value of Y for this sysparm. A value of Y for this sysparm will supersede sysparm UCC1EXPR.

This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025".

UNCATARC*n*

This sysparm determines the catalog action to be taken for any archive tape dropped (deleted) in the index maintenance functions (merge/purge/delete) processing. The default value of N causes SAMS:Disk to leave the catalog status unaltered. A value of Y causes SAMS:Disk to uncatalog any tape that is deleted from the archives. This sysparm is intended for joint use with sysparm ARCTNAME.

UNCATDELn

If you let sysparm UNCATDEL default to a value of N, SAMS:Disk treats DSNINDEX Julian expiration date "99000" as a never scratch date. This is equivalent to the other "never scratch" dates of 99365, 99366 and 99999.

But if you set sysparm UNCATDEL to a value of Y, SAMS:Disk treats Julian expiration date "99000" as a "catalog control date." That is, data sets backed up or archived with an expiration date of 99000 will be deleted automatically by the SAMS:Disk function IXMAINT when the data sets become uncataloged. What this means is, if your catalog goes bad or someone uncatalogs some data sets, that all backups and archives of these data sets will be deleted on the next IXMAINT run. Although you may use EXPDT=99000 for any purpose, it was intended for use with archiving generation data groups (GDGs), the generations of which are managed by the catalog. For example, to archive all GDG generations, leaving the two most recent on disk, deleting generations that roll off of the catalog, specify sysparm UNCATDEL with a value of Y and:

```
SCAN REALVOLS SELECT CRITERIA=(GDGGEN,LE,-2) ARCHIVE RECAT,EXPDT=99000
```

UNCATFDRn

If this sysparm is specified with the default value of N, SAMS:Disk does not do additional catalog checking during UNCATPSUy processing. This avoids the additional overhead caused by the catalog checking. If this sysparm is specified with a value of Y, SAMS:Disk does additional catalog checking prior to deleting a catalog entry during UNCATPSUy processing. The additional catalog checking is in the form of a catalog locate to determine if the catalog entry may have been created by FDRABR. Once the catalog locate has been done, the first byte of DSCB TTR information is tested for a x'80' value. If the test is true, SAMS:Disk assumes the entry was created by FDRABR and does not uncatalog the data set.

UNCATPSUn

If you are recataloging archived data sets to the SAMS:Disk pseudo-volume, or if archived data sets are being left cataloged to permit auto-restores, specify a value of Y for this sysparm to remove obsolete catalog entries during merge, index maintenance, or ixupdate processing when the last DSNINDEX record is deleted with that data set name. SAMS:Disk uncatalogs data sets from the pseudo-volume only if the DSNINDEX record being deleted is an archive copy (that is, the data set was scratched at the time the copy was made), and a value of R is specified for this sysparm.

When this sysparm is specified with a value of Y, SAMS:Disk determines if this is an "obsolete" catalog entry by verifying that this is the last DSNINDEX entry for the specific data set being deleted (that is, this copy of the data set has expired and no other copies exist). At that time, it checks to see if the data set is still cataloged. If so, the following actions are taken:

- a. If it is cataloged to the SAMS:Disk pseudo-volume, SAMS:Disk uncatalogs it.

- b. If it is still cataloged to its original source volume or to the “staging volume” being used for auto-restores, SAMS:Disk issues an OBTAIN macro to see if the data set is still there. It may have been restored or re-created since it was archived and scratched). If the data set is found, the catalog is left unchanged; otherwise the entry is deleted.
- c. If it is cataloged to a volume other than those mentioned above, no further processing is done. The catalog entry is left unchanged. For sysparm UNCATPSU, the value will no longer be internally set for auto-restore. When the sysparm is set to a value of Y, the catalog entry for a data set will be deleted in the event a DSNINDEX does not exist for the data set. If the sysparm is set to its default value of N, the catalog entry will not be deleted.

Care should be taken when using this sysparm in a shop where offline production volumes exist. For if sysparm IXMVOLONn (which is the default), is also specified the OBTAIN will fail for data sets cataloged to offline volumes. These data sets will be uncataloged. Please refer to sysparm *IXMVOLON* on page 155 for more information.

If this sysparm was specified with a value of R, SAMS:Disk follows the above steps only if the DSNINDEX record being deleted is the LAST ARCHIVE copy in this files data set. No uncatalog action is done if this is a backup copy, even if it is the last DSNINDEX record of this data set name in the files data set.

If this sysparm is specified with a value of X, all catalog actions for processed data sets will be bypassed during IXMAINT. This will leave the cataloged entries intact for any subsequent REBUILD processing.

UNLOADEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the UNLOADEX user exit. Turn to the user exit description for *UNLOADEX* on page 295 for more information.

UNMOVEARn

The default value of N for this sysparm prevents SAMS:Disk from moving unmoveable data sets during archive/backup and restore processing. SAMS:Disk does not officially support unmoveable data sets and therefore cannot guarantee the integrity of the data set after processing it. To permit archive/backup and restore processing of unmoveable data sets, specify this sysparm with a value of Y.

UNMOVEVCn

The default value of N for this sysparm prevents SAMS:Disk from moving unmoveable data sets using the Move/Copy function. SAMS:Disk does not officially support unmoveable data sets and therefore cannot guarantee the integrity of the data set after processing it. To move unmoveable data sets using the Move/Copy function, specify this sysparm with a value of Y.

USERSENQmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the USERSENQ user exit. Turn to the user exit description for *USERSENQ* on page [296](#) for more information.

To perform its intended task, the SAMS:Disk auto-restore function internally assigns and uses value “DMSNOENQ” for this sysparm.

USERVENQmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the USERVENQ user exit. This exit is used to monitor and/or update the allocation type (DISP=SHR or OLD) SAMS:Disk will use to allocate VSAM data sets. Turn to the user exit description for *USERVENQ* on page [297](#) for more information.

USRCLENxx

This sysparm specifies the LEN of the user code in the DSCB. The default value is 12. This sysparm is used in DASD Space Billing.

USRCODEXmmmmmmmmmm

This sysparm specifies the module name to be invoked for the USRCODEX user exit. Turn to the user exit description for *USRCODEX* on page [298](#) for more information.

VCBYPASSn

The default value of this sysparm is N, which suppresses printing of messages stating the reason why the Move/Copy function bypassed a data set for migration. To cause such messages to be printed, specify a value of Y. Note: The COPY command will always print a message stating the reason why it bypassed a data set.

VCCHGBITn

The default value of this sysparm is N, which causes SAMS:Disk, when it moves a data set, to leave the change bit the same as it was before the data set was moved. To cause SAMS:Disk to set the change bit on each moved data set, specify a value of Y. Setting the change bit on each moved data set may increase the number of data sets backed up in implicit archive runs, but may simplify the recovery of packs. See also sysparm *RSCHGBIT* on page [181](#) in this manual.

VCCOPYEXmmmmmmmmmm

This sysparm specifies the module that is given control by the Move/Copy function just prior to copying an explicitly named data set (using the COPY command). All selection screening by SAMS:Disk has been completed and the user's exit module is given a chance to override the copying. A return code of 2 from the exit causes SAMS:Disk to bypass the data set. Turn to the user exit description for *VCCOPYEX* on page [299](#) for more information.

VCDMIGEXmmmmmmmmmm

This sysparm specifies the module that is given control by the Move/Copy function just prior to the migration of a data set (using the MIGRATE command). All selection screening by SAMS:Disk has been completed and the user's exit module is given a chance to override the migration. A return code of 2 from the exit causes SAMS:Disk to bypass the data set. Turn to the user exit description for *VCDMIGEX* on page 300 for more information.

VCUDSORGn

The default value of N for this sysparm causes the Move/Copy function to bypass data sets with unknown organization types. If you want the Volume Configurator to treat such data sets as sequential data sets, specify a value of Y.

VOLSELREn

Specifying a value of Y for this sysparm will require the user to code a VOLUME= parameter on all SELECT statements. The default value of N is equivalent to coding VOLUME=/ on a SELECT statement which causes SAMS:Disk to attempt to process ALL data sets selected by the SELECT command. This can allow unwanted processing to occur. For example, if a user unintentionally runs:

```
SCAN REALVOLS
  SELECT CRITERIA=(DSORG,EQ,VSAM)
  DELETE
```

Note: For other related sysparms, refer to the descriptions of SELECTRE (page 184 in this manual) and DSNSELRE (page 145 in this manual).

VSACCESSc

The default value for this sysparm is "C", which instructs SAMS:Disk to read VSAM clusters with control interval access. Specify this sysparm with a value of "L" to force SAMS:Disk to read VSAM clusters using logical access methods. When specifying this sysparm, also look at the value specified for sysparm VSARCFMT.

VSAM clusters that do not observe logical record definitions can only be read with control interval access. SAMS:Disk automatically detects these data sets, and reads them using control interval access, regardless of the value specified in this sysparm.

VSAM clusters that have been compressed with SOFTWARE's VDC product can only be read with logical access methods. SAMS:Disk automatically detects these data sets, and reads them using logical access methods, regardless of the value specified in this sysparm.

RACF users should note that using control interval access requires having CONTROL authority for the VSAM clusters. This is higher than UPDATE authority, which is higher than READ authority required by logical access methods. If it is unacceptable to require CONTROL access to read VSAM clusters, specify this sysparm with a value of "L".

Note: If the CIARPT parameter on the REPORT command is present in the same SYSIN as the BACKUP or ARCHIVE command, control access is required and VSACCESSC will be used.

VSACESEXmmmmmmmmmm

This sysparm specifies the name of the user exit that is given control prior to any read of a VSAM cluster. The purpose of the user exit is to interrogate (and change if desired) both the access type that SAMS:Disk will use on a given data set, and also the format in which to write the data set when archiving it either to tape or disk. Turn to the user exit description for VSACESEX on page 301 for more information.

VSAIXDEFy

This sysparm is used during VSAM restore and recover processing to determine if the alternate indexes that were associated with a VSAM cluster are to be defined after the base cluster is restored by SAMS:Disk. The default value of Y causes the indexes to be defined. Specify a value of N to cause SAMS:Disk to only restore the base cluster, without redefining the alternate indexes. This sysparm can be overridden on the RESTORE and RECOVER commands by specifying the parameters AIXDEF or NOAIXDEF. Refer to the topic *"Parameters for VSAM Clusters"* on page 240 in the *User's Guide* for further information regarding these parameters. See also sysparm VSBLDINX on page 202 for information regarding building the alternate index after it is defined.

VSALLOEXmmmmmmmmmm

This sysparm specifies the module that is given control prior to all VSAM defines in SAMS:Disk. The purpose of the user exit is to interrogate (and change if desired) the primary and secondary space allocation amounts to be used in the define. Turn to the user exit description for VSALLOEX on page 303 for more information.

VSALOSHRn

This sysparm controls the disposition used for allocating VSAM data sets. The default value is N, which causes SAMS:Disk to allocate only those data sets defined with share options (1 3) with DISP=SHR, and all others with DISP=OLD. To cause SAMS:Disk to use DISP=SHR for all VSAM data set allocations, specify a value of Y. Note that specifying a value of Y can cause an integrity exposure since SAMS:Disk does no ENQ for VSAM resources.

To perform its intended task, the SAMS:Disk auto-restore function internally assigns and uses a value of "Y" for this sysparm, which causes DISP=SHR to be used for all VSAM data set allocations.

VSAMSUPPn

The default value for this sysparm is N, which indicates to SAMS:Disk that the VSAM support is not active. Specify a value of Y to activate the VSAM support.

VSAPPENDn

This sysparm is used during restore and recover of VSAM data sets. Its purpose is to generate new data and index component names for those clusters originally defined with VSAM-generated component names based on timestamp information. Data and index components that were given explicit names when originally defined are not affected by this sysparm.

If a value of Y is specified, SAMS:Disk generates each new name (to replace the timestamp version) using the base cluster's data set name, with ".DATA" appended for the data component, and ".INDEX" appended for the index component.

For example, if a base cluster of LABS.TJP.PMRFILE is being restored, the data component's name will be LABS.TJP.PMRFILE.DATA and the index component's name will be LABS.TJP.PMRFILE.INDEX. If the base cluster's name is too long to add the suffix (.DATA or .INDEX), SAMS:Disk will let VSAM generate a new name based on timestamp information.

Two other sysparms, VSDATSUF and VSINXSUF, can also be used for this same purpose, with or without specifying VSAPPEND. They control the new name for each component independently, and require you to specify the suffix to be appended. Therefore they are normally used only if suffixes other than .DATA and .INDEX are desired.

If the default value of N is in effect, and neither VSDATSUF nor VSINXSUF is specified, SAMS:Disk will instruct VSAM to generate another component name based on timestamp information.

See sysparms VSDATSUF and VSINXSUF on pages [203](#) and [204](#) respectively.

VSARCAIXn

When this sysparm is specified with the default value of N for an archive or backup run, SAMS:Disk bypasses processing alternate indexes and clusters as separate entries. This sysparm has no effect for an explicit archive or backup of a base cluster.

When specified with a value of Y for an implicit backup run, SAMS:Disk processes the alternate indexes as separate data sets. And, entries are placed in the files data set for the alternate indexes. This sysparm must be specified with a value of Y for an explicit archive or backup of the alternate index.

Note: This sysparm is ignored for an implicit archive run. This is because all AIX associations and DEFINE information is saved with the base cluster before the delete takes place, not as a separate data set.

VSARCFMTI

This sysparm controls the format of archive data for VSAM, if VSACCESSc has been specified. If VSACCESSl is specified, this sysparm is not considered and the data is archived in the logical record format. When the default value of L is specified, logical records are written to the archive medium. Specify a value of C to force

VSAM clusters that do not observe logical record definitions cannot use logical formats. These VSAM data sets are automatically detected by SAMS:Disk and formatted in control interval images, regardless of the value specified in this sysparm.

VSAM data sets that have been compressed with SOFTWARE's VDC product can only use logical formats. SAMS:Disk automatically detects these data sets, and formats them in logical records, regardless of the value specified in this sysparm.

When a control interval image format is used, the control interval size cannot be changed when the data set is restored.

This sysparm specifies the module that is given control prior to all VSAM defines in SAMS:Disk. The purpose of the user exit is to interrogate (and change if desired) the BUFFERSPACE AND CISZ values to be used in the define. In addition, certain data set attributes can be changed by this exit: RECOVERY vs. SPEED, UNIQUE vs. SUBALLOC, REUSABLE vs. NONREUSABLE, and data set SHARE OPTIONS. Turn to the user exit description for *VSATTREX* on page 304 for more information.

The default value of this sysparm is N, which causes SAMS:Disk to use the original CISZ and BUFFERSPACE values found on the archive copy when processing data sets. By specifying a value of Y, all data sets that are defined by SAMS:Disk in the job step will have CISZ and BUFFERSPACE calculated by AMS. These values may or may not correspond to the original values with which the cluster was defined.

This sysparm should not be used when processing ESDS clusters. Therefore, SAMS:Disk automatically detects these data sets and will not modify the CISZ for them.

Note: This sysparm should not be used if AIM data bases are restored by SAMS:Disk. If the CISZ is changed inadvertently, the data base becomes inoperable. It must be restored with its original CISZ!

Specify the name of the catalog into which the workfiles are to be defined to build alternate indexes when restoring VSAM clusters. If this sysparm is not specified, the catalog will be determined by the following order: (1) any user catalog specified in a STEPCAT or JOBCAT dd statement, or (2) if the data set names on the KQCUT1 and KQCUT2 dd statements are identified with qualified entry names and the first qualifier is the same as the name of a user catalog or the alias of a user catalog, the user catalog so identified is used, or (3) the master catalog is used.

VSBIXPSWnnnnnnnn

If the catalog specified in the VSBIXCAT sysparm is password-protected, specify this sysparm with the catalog's update (or higher-level) password. If this sysparm is not specified with the password, the operator may be prompted to supply the correct password.

VSBLDINXy

This sysparm is used during VSAM restore and recover processing to determine if alternate indexes are to be built automatically after they are defined by SAMS:Disk. The default value of Y causes SAMS:Disk to build the indexes. Specify a value of N to cause SAMS:Disk to only define the alternate index without actually rebuilding it. This sysparm can be overridden on the RESTORE and RECOVER commands by specifying the parameters BLDINX or NOBLDINX. Refer to the topic *"Parameters for VSAM Clusters"* on page 240 in the *User's Guide* for further information regarding these parameters.

VSCATMAXnnnn

Specify a 4-digit number at least one greater than the number of user catalogs defined in the master catalog. This value is used to build a dynamic table of user catalog names in the implicit VSAM support. Note that this table is only built when sysparm VSCATPTN is specified as Y (or C) and a catalog data set name pattern is specified (as opposed to an explicit catalog name) on the CATALOG= parameter of the SCAN command. The default value for this sysparm is 0050.

VSCATPTNy

The default value for this sysparm is Y, which allows the user to enter a catalog data set name pattern (along with explicit names) on the CATALOG= parameter of the SCAN command in the implicit VSAM support. Specify a value of N to disallow this feature.

If a value of Y (or C) is specified and the user specifies a catalog data set name pattern, the master catalog is searched for all user catalogs defined in it. These user catalogs, along with the master catalog, are compared to the specified pattern for a match. All catalog names that match will be processed as if they had been explicitly named on the CATALOG parameter. Note: This feature does incur the additional overhead required to search the master catalog.

When a value of Y is specified and a catalog pattern is entered on the SCAN CAT= command, SAMS:Disk will verify that at least one catalog matches the pattern specified. If none match, processing is terminated and none of the commands in the job stream are executed. If you wish to fail only this SCAN command and allow any others to complete, specify a value of C for this sysparm. This causes SAMS:Disk to continue processing other commands even though an invalid catalog pattern name was specified.

VSCIACATy

The default value of this sysparm is Y, which causes catalog information to be printed on the cluster internal analysis report (CIA) of the VSAM support. Specify-

ing a value of N causes the information to be suppressed; it can then only be printed by explicit specification on the CIAOPT= parameter.

VSCIADETn

The default value of this sysparm is N, which causes the graphic analysis section of the cluster internal analysis report (CIA) of the VSAM support to be suppressed. Specify a value of Y to have the graphic analysis section of the report printed.

VSCOMPNMn

The default value of this sysparm is N, which suppresses printing of the data set name associated with the data and index components on the Cluster Data Report of VSAM. Specify a value of Y to cause SAMS:Disk to print the data set name in this report. If this parameter is specified in the job stream, it must be specified in the PRINT step of the job.

VSCRIMAXnnnn

Specify as “nnnn” the maximum number of subparameters to be allowed on the SELECT and EXCLUDE commands for the implicit VSAM support. The default value is 0150. Normally the default value should not require any change. The maximum value that may be specified is 0255.

VSDATEFTn

The default value of this sysparm is N, which causes SAMS:Disk to use the Fujitsu formats and names supporting 4-digit year numbers for VSAM Restore and Recover. Specify a value of Y if it is determined that your system does not support the default date formats.

VSDATSUFxxxxxxxx

This sysparm specifies a 1- to 8-character suffix to be used by SAMS:Disk at Restore time or in the Move/Copy function at COPY newname time to generate new data component names for those clusters that have VSAM-generated component names based on the timestamp. The sysparm defaults to a value of blanks, such that the newname function is inactive. To activate the newname function, specify a non-blank value for this sysparm or specify sysparm VSAPPEND with a value of Y. If both sysparms are specified, VSDATSUF will override VSAPPEND.

For example, if this sysparm is specified with a value of D, and cluster LABS.TJP.XREFFILE is being restored, the data component will be named LABS.TJP.XREFFILE.D.

Note: Specifying sysparm VSAPPEND with a value of Y is the same as specifying this sysparm with a value of DATA and sysparm VSINXSUF with a value of INDEX.

VSDEFCATy

The default value of this sysparm is Y, which allows catalog management to determine the catalog in which to put the cluster, based on Fujitsu-documented catalog

selection criteria. Specify a value of N to cause SAMS:Disk to not let catalog management determine the catalog. Instead, SAMS:Disk will define the cluster in the same catalog from which it was archived. If this catalog is not the same one that catalog management would have used, a STEPCAT dd statement for the catalog will probably be needed also. Note: This sysparm has no effect when the catalog is explicitly named on the RESTORE or RECOVER command.

VSDSPACE*n*

This sysparm lets you specify if sub-allocated VSAM data sets in VSAM data spaces should be bypassed for processing using the DSCL SCAN REALVOLS command. This sysparm is only relevant for the DSCL SCAN REALVOLS command, and not the DSCL SCAN CATALOG command. Processing the sub-allocated VSAM data sets incurs extra overhead for SCAN REALVOLS, due to additional catalog processing.

When this sysparm is specified with the default value of N, SAMS:Disk processing bypasses sub-allocated VSAM data sets when using the DSCL SCAN REALVOLS command.

When this sysparm is specified with a value of Y, SAMS:Disk processing allows selection of sub-allocated VSAM data sets when using the DSCL SCAN REALVOLS command.

VSIDCUTP*xxxxxxxx*

If desired, specify a one- to eight-character prefix to be used by SAMS:Disk if it needs to generate data set names for the KQCUT1 and KQCUT2 dd statements during restore or recover operations. If a value is not specified, the high-level qualifier of the first alternate index being built will be used. For more details, refer to the topic *"Activating VSAM Support"* on page 27 of the *Installation Guide*.

VSIDCUTV*xxxxxxxx*

If you do not specify the KQCUT1 and KQCUT2 dd statements directly in your restore or recover JCL, you must specify this sysparm with the volume(s) that SAMS:Disk can use for the sort work data sets required by BLDINDEX processing. For more details, refer to the topic *"Activating VSAM Support"* on page 27 of the *Installation Guide*. Specify from 1 to 10 volumes in the list, all of which must be the same device type.

VSINXSUF*xxxxxxxx*

This sysparm specifies a 1- to 8-character suffix to be used by SAMS:Disk at Restore and in the Move/Copy function at COPY newname time to generate new index component names for those clusters that have VSAM-generated component names based on the timestamp. The sysparm defaults to a value of blanks, such that the newname function is inactive. To activate the newname function, specify this sysparm with a non-blank value, or specify sysparm VSAPPEND with a value of Y. If both sysparms are specified, the value specified for this sysparm will override VSAPPEND.

For example, if this sysparm is specified with a value of INX, and cluster LABS.TJP.XREFFILE is being restored, the index component will be named LABS.TJP.XREFFILE.INX.

Note: Specifying sysparm VSAPPEND with a value of Y is the same as specifying this sysparm with a value of INDEX and sysparm VSDATSUF with a value of DATA.

VSMAFTEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the VSMAFTEX user exit. Turn to the user exit description for *VSMAFTEX* on page [306](#) for more information.

VSMPRIEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the VSMPRIEX user exit. Turn to the user exit description for *VSMPRIEX* on page [307](#) for more information.

VSMPURGEy

The default value of this sysparm is Y, which causes SAMS:Disk to delete unexpired data sets. Specify a value of N to instruct SAMS:Disk to not allow deletion of unexpired data sets. The default value of Y is equivalent to the use of the PURGE parameter when issuing a DELETE via Access Method Services.

VSMSGEMPn

The default value of this sysparm is N, which causes SAMS:Disk to bypass issuing message 2517 for empty VSAM clusters that are being archived. Specify a value of Y to have the message be issued whenever the condition is encountered. For more information, refer to the message description for 2517 on page [169](#) of the *Message Manual*.

VSMSGTOTn

The default value of this sysparm is N, which causes SAMS:Disk to bypass issuing message 2562 when a difference in record totals is detected between SAMS:Disk processing and the catalog. Specify a value of Y to have the message issued whenever the condition is encountered. For more information, refer to the message description for 2562 on page [184](#) of the *Message Manual*. Note that message 2832 might not be issued, depending on the value specified for sysparm VSMSGWRN.

VSMSGWRNn

The default value of this sysparm is N, which causes SAMS:Disk to bypass issuing message 2832 when the SAMS:Disk record total is less than the value found in the catalog for the data set. Specify a value of Y to have the message issued whenever the condition is encountered. For more information, refer to the message description for 2832 on page [215](#) of the *Message Manual*.

VSMSPWEXmmmmmmmmmm

This sysparm specifies the module that is invoked during initialization processing in the VSAM support. The purpose of the module is to allow the user to change the default SAMS:Disk master password (currently implemented only in the VSAM support). Turn to the user exit description for *VSMSPWEX* on page 309 for more information.

VSMSTATSn

The default value of this sysparm is N, which causes SAMS:Disk to bypass printing of additional data and index component statistics on the Cluster Detail Analysis report. Specify a value of Y to force printing of these additional statistics. The statistics for each component will be printed on an additional print line. Included with these statistics are the component's high used and high allocated RBAs, along with tracks allocated. Note: If this parameter is overridden in the user's job stream, the override must be specified in the PRINT step.

VSNEWNEXmmmmmmmmmm

This sysparm specifies the module that is given control prior to all VSAM defines in SAMS:Disk. The purpose of the user exit is to interrogate (and change if desired) the data set name with which the component will be defined. The exit can either change a name that SAMS:Disk will use, or it can instruct SAMS:Disk to let AMS generate a name for the component. Turn to the user exit description for *VSNEWNEX* on page 310 for more information.

VSNOERASn

The default value of this sysparm is N, which causes SAMS:Disk to honor the erase option when deleting a VSAM cluster. Specify a value of Y to cause SAMS:Disk to override the erase option when deleting a VSAM cluster. If SAMS:Disk attempts to delete a cluster that was defined with the erase option and an error prevents its deletion, this sysparm will cause SAMS:Disk to reattempt the delete with a no erase option. This causes the cluster to be deleted from the catalog without overwriting the data component with binary zeroes. Note that even if this sysparm is specified as a Y, the default erase/no erase value will be attempted first. The delete with no erase will only be attempted if the first delete fails.

VSNOSCROn

The default value for this sysparm is N, which causes SAMS:Disk not to honor the SCRATCH option if it has been specified and the delete fails. This would most likely occur when a data sets catalog entry is pointing to an unavailable volume.

Specify a value of Y to cause SAMS:Disk to retry a delete that failed. The second attempt to delete will be performed with the NO SCRATCH option specified. This causes the entry to be deleted from the catalog (BCS or basic catalog structure)

without trying to access the volume. SAMS:Disk will only retry a SCRATCH for the following KQC3009I return codes:

Return Code	Reason Code	Meaning
8	214	Unit(s) not available
8	218	Specified volume is not mounted
50	18	I/O error occurred while processing a VVDS
102	5	I/O error occurred while processing a VVDS

Typically this sysparm would be set to Y when restoring a VSAM data set that has a cataloged entry pointing to a corrupted VVDS, and you want to restore to a different volume. Issuing a standard delete (either by SAMS:Disk or KQCAMS) will fail because the BCS still contains an entry for the data set. When this occurs, the NOSCRATCH option is required to clean up the catalog entry without requiring access to an unavailable volume.

We recommend that you use the default value of this sysparm (N) for normal SAMS:Disk processing, and only specify a value of Y during a volume recovery situation. Note that even if this sysparm is specified as Y, a standard delete request will be attempted first; the delete with NOSCRATCH will be attempted only if the first delete fails with a specific set of reason codes.

VSONLINE

This sysparm controls the amount of checking done to determine if the volumes containing VSAM clusters are online to the system. It applies only to the implicit mode of processing VSAM data sets.

Since removable volumes and offline volumes (or even obsolete/garbage entries pointing to currently invalid volumes) are generally exceptions rather than the rule, the default value of this sysparm is N. SAMS:Disk will avoid the overhead of checking, and assume that needed volumes (the volumes associated with each cluster in the catalog) are available. Mount requests may occur if some are not available. This may be desirable if you know a few volumes are offline, but you want them mounted for backup processing, for example.

Specify a value of Y if you want SAMS:Disk to verify that needed volumes are online. This will prevent unexpected mount requests for volumes that are not available. Verification is done by dynamically allocating the cluster with the “NOMOUNT” attribute. If the allocation of the cluster fails, processing for cluster will be bypassed.

The Y option will eliminate unexpected mount requests for VSAM data sets residing on offline volumes, but unusual catalog conditions and entries may cause other unexpected mounts before SAMS:Disk gains control. These are usually exception

cases. Examine the catalog entry, the alias pointers in the master catalog, and other relevant information to determine and correct the condition causing the unexpected mount request.

Note: The verification done as a result of this sysparm being set to a value of Y may increase the run time for your job. It may also increase the number of messages written to your job log. Adjusting the value of the MSGLEVEL parameter on your JOB card may reduce the number of messages.

VSPREDEFn

The default value of this sysparm is N, which will cause SAMS:Disk to not allow restore or recover commands to overwrite preallocated VSAM clusters. Specify a value of Y to allow a restore to overwrite preallocated VSAM clusters with the REUSE attribute. Note that SAMS:Disk cannot be held responsible for the integrity of the restored data when a value of Y is specified.

You should not specify a value of Y during VSAM volume recover after an EDF-Catalog recovery when the information in the VVDS is not correct.

See sysparm PREALLOC on page [163](#) for restore of non-VSAM data sets.

(To protect integrity, the SAMS:Disk auto-restore function internally assigns and uses a value of “N” for this sysparm, which prevents VSAM data sets being auto-restored into existing clusters.)

VSRESCLDn

The default value for this sysparm is N, which causes SAMS:Disk to suppress dynamic printing the VSAM cluster detail report. Specify a value of Y to cause SAMS:Disk to print a cluster detail report dynamically whenever a VSAM cluster is recovered or restored. This report will be printed on the restore report that is normally produced at restore time. The lines that are printed are based on the values specified for sysparm RPTCLSEL.

VSSEQCNTnnn

This sysparm allows you to specify a numeric quantity from 1 to 999 to indicate the number of “key out of sequence” errors SAMS:Disk should tolerate (per data set) at restore time before terminating the restore for that data set. If the default value of 0 is used, the first such error will cause the restore to be terminated for that data set. If you specify a value greater than 0, and the number of errors is less than the value specified, the data set will be restored but the records that were out of sequence must be loaded manually back into the cluster. A dump of every record that is out of sequence will be printed to the MSGPRINT error report for user inspection.

This sysparm will normally be required only if you have used sysparm VSSEQERR to allow a VSAM cluster with such problems to be archived.

VSSEQERRy

This sysparm has a default value of Y, which causes SAMS:Disk to stop processing a VSAM cluster if an out-of-sequence key is encountered. Specify a value of N to cause SAMS:Disk to ignore out-of-sequence keys in a key-sequence data set and to continue processing the data set. Message 2839 will be issued regardless of the value specified for this sysparm. If a value of N is specified, no records will be lost on the archive medium but the data set may have to be restored by non-conventional means (such as extract, sort, and then using the REPRO facility of KQCAMS) since the keys are not in sequence.

VSSETUSEn

This sysparm controls the way the last use date is updated in the format-1 DSCB when VSAM clusters defined in an EDF-Catalog are restored. The default value of N causes the last use date to be updated according to Fujitsu rules for EDF-Catalogs.

Specifying a value of Y causes SAMS:Disk to set the last use date as follows: If the last use date was non-zero when the data set was archived, SAMS:Disk will use that date when it is restored. If the last use date was zero when the data set was archived, SAMS:Disk will set the last use date to equal the creation date of the data set when it is restored.

VSSPEEDOy

The default value of this sysparm is Y, which causes SAMS:Disk to use the “speed” option, regardless of the value with which the cluster was defined. Specify a value of N to cause SAMS:Disk to use the “speed” or “recovery” option with which the cluster was defined.

VSSPUSEDy

The default value of this sysparm is Y, which causes SAMS:Disk to consolidate secondary extents in the original cluster and increase the primary allocation (when required) accordingly. This results in defragmentation of the data set. However, if the pack is badly fragmented already, the larger amount of space may not be available for the primary request. In this case, specify a value of N to cause SAMS:Disk to use the original allocation amounts for all DEFINE requests.

VSSTUMPLnnnnnn

This sysparm allows you to specify the number of bytes required to hold the largest catalog entry returned from a catalog locate. The default value is 026000. Specify this sysparm only in response to message 2561.

VSSYSVERn

Normal OPEN processing of a VSAM data set in this environment will cause a VERIFY to be issued against any cluster with an out-of-sync end-of- file marker. This is internal to OPEN processing and causes the VERIFY to be issued prior to the application program receiving control back from OPEN. This sysparm’s default

of N turns off this automatic VERIFY feature. Specify a value of Y to allow the automatic VERIFY to take place.

VSUSEBLKn

The default value of this sysparm is N, which causes SAMS:Disk to use the control interval size (CISZ) of the data component for selection criteria BLKSZ and to be printed under the BLKSZ header in the VSAM MVD and SVD reports. Specify a value of Y to use the cluster's physical block size instead of the CISZ.

VSUNOPENn

The default value of this sysparm is N, which causes SAMS:Disk to ignore a defined but never opened VSAM cluster when the criteria (MODIFIED,EQ,YES) is specified. Specifying a value of Y forces SAMS:Disk to select such clusters once, but never again until it is actually modified.

There is one exception, and that occurs on the day the cluster is defined. On that day, SAMS:Disk will select "defined, but never opened" clusters every time the criteria (MODIFIED,EQ,YES) is executed.

The standard way to detect if a VSAM cluster has been modified is reviewing the TIMESTAMP information and/or the DS1DSCHA (Change bit) in the format-1 DSCB. AMS sets neither of these for a never-opened cluster. So, when the TIMESTAMP is zero, SAMS:Disk compares the CREDIT to the ARCDT. If CREDIT is greater than or equal to the ARCDT, SAMS:Disk considers the cluster modified.

Specifying VSUNOPEN with a value of Y ensures that a "defined, but never opened" cluster is backed up when the criteria MODIFIED,EQ,YES is executed.

VSUSEMODn

The default value for this sysparm is N, which causes SAMS:Disk to print the last use date from the format-1 DSCB for the data component of a VSAM cluster under the USEDTE heading in the VSAM MVD and SVD reports. For a standard VSAM catalog data set using the default value, blanks are printed in this column. Specify a value of Y to cause a VSAM cluster's last modified date to be printed under the USEDTE heading.

VSVERIFYn

The default value for this sysparm is N, which causes SAMS:Disk to bypass processing of a cluster when the need to verify it is detected at OPEN time. The cluster must be verified manually. Specify a value of Y to instruct SAMS:Disk to issue a VERIFY when the need is detected at OPEN time. If the VERIFY succeeds, SAMS:Disk will then attempt to OPEN the data set again and continue processing.

VTOCREADe

This sysparm controls which of three VTOC readers SAMS:Disk will use. Although these VTOC readers are functionally equivalent, they perform quite differently. The default value is E.

Table 4-11. Vtoc Reader Names and Descriptions

Reader	Description
E	Causes SAMS:Disk to use EXCP processing and is quite fast.
B	Causes SAMS:Disk to use the BSAM reader. This is a standard Fujitsu facility, but is quite slow. This can increase SAMS:Disk processing time up to 200 percent on some report writers.
O	Causes SAMS:Disk to use the OBTAIN macro VTOC reader. This also is a standard Fujitsu facility and is even slower than option B. This can increase SAMS:Disk processing time up to 1100 percent on some report writers.

Use options B or O only when VTOC problems prohibit the use of the SAMS:Disk EXCP VTOC reader.

WTORESRLy

The default value of this sysparm is Y, which causes SAMS:Disk to issue a WTO message for each RESERVE issued on a pack during PDS Compress in a shared-DASD environment. Specify a value of N to cause SAMS:Disk to not issue a WTO message under these circumstances.

XCPARCEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the XCPARCEX user exit. Turn to the user exit description for *XCPARACEX* on page [311](#) for more information.

XCPDSNEXmmmmmmmmmm

Specify this sysparm to provide the module name to be invoked for the XCPDSNEX user exit. Turn to the user exit description for *XCPDSNEX* on page [312](#) for more information.

Chapter 5. User Exits

General Information

Modules may be incorporated in SAMS:Disk to satisfy unique processing requirements of an installation. These modules are identified to SAMS:Disk by appropriate system parameter entries in member SYSPARMS of the parmlib data set. During execution of SAMS:Disk, the exit modules are dynamically loaded when they are called for the first time.

User exit modules may directly invoke SAMS:Disk functions by use of the MI dynamic interface, as illustrated in the topic *"Module Call Interface"* on page [355](#) in this manual. User exits using the MI dynamic interface must be link-edited with the MI interface module from the SAMS:Disk library. Instructions for doing this are under the *"Assembly and Link-Edit of SAMS:Disk Modules"* topic on page [353](#) in this manual.

If your exit is written in COBOL or PL1 rather than assembler, and calls other routines, it is recommended that you use the MI2 interface rather than MI (fast path). See *"MI Fast Path"* on page [326](#) for more information about MI2.

Note: User exits may need to be re-assembled each time you upgrade your SAMS:Disk release level (i.e., source code to ADSMI002 has changed).

In order to provide compatibility for existing SAMS:Disk modules with 24-bit addressing, only the following AMODE/RMODE combinations are valid for user exits:

```
AMODE 24, RMODE 24
AMODE 31, RMODE 24
AMODE ANY, RMODE 24
```

Each of the user exits is discussed under the sysparm entry name that is necessary to define the respective module to SAMS:Disk for invocation.

Assembler language conventions are followed in passing parameters to the exit module. Register 1 will point to a list of addresses, each of which points to a specific parameter as defined for the exit. In most cases, the exit module must return a halfword code via the designated parameter to indicate to SAMS:Disk what further actions are to be taken.

If you use the Common File Handler Interface (described on page [30](#) in this manual), and you are not already executing a SAMS:Disk function, and files data set logging is activated, the function you are running needs to be stumped within your user exit prior to any file updates. This is necessary for files data set logging as the current function is part of the information captured during logging.

To implement this, add the following to your user exit (this call need only be executed once):

```
* Stump the function to retrieve during files data set logging
#CALL ADSUT077, ($ISRT,FUNCTION,CTYPE,K08,USRFUNC,UT077RC),VL
*
$ISRT      DC CL5'$ISRT'      Service = Insert
FUNCTION   DC CL8'FUNCTION'   Field name for retrieval
CTYPE      DC CL2'C'          Service type = character data
K08        DC H'8'            Length of field
USRFUNC    DC CL8'            ' Fill in user function name
UT077RC    DC H'0'            Return code
```

Figure 5-1. Code to Stump User Exit Functions

Using Previous Releases of User Exits

Note: If you are upgrading from an 8.2 release of SAMS:Disk, you can ignore this section.

User exits built for previous releases of SAMS:Disk are compatible with release 9.0 with one exception:

1. If the SAMS:Disk provided PROC macro is used to begin the exit program, verify whether or not the PROC macro is specified with an `OPTIONS=(blank)` parameter. For example:

```
csectname PROC OPTIONS=
```

If this `OPTIONS=` parameter (without a value) is found in any user exit, that exit must be reassembled.

Note: Modifications to the user exit are not necessary, reassembly alone will use an updated version of the PROC macro provided with release 9.0, thus creating a release 9.0 compatible module.

Use of PROC statements with other values are all compatible with release 9.0 and can be moved directly into the 9.0 libraries. All user exits using standard assembly entry and exit code are also fully compatible with release 9.0. For example, typical uses of the PROC macro that generate code compatible with release 9.0 are:

```
csectname PROC OPTIONS=STATIC
csectname PROC OPTIONS=NOSAVE
csectname PROC OPTIONS=MAIN
csectname PROC
```

Finally, if you have not correctly implemented the PROC macro, attempting to invoke a user exit from SAMS:Disk will result in User Abend 1547-16 followed by:

MTF005I OBSOLETE MIVT SERVICE REQUESTED

User Exits Cross Reference Matrix

Exit Name	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD	XCOPY
ALLOEXIT				•	•		•															
ALOCATEX				•	•		•											•				
ALOPOSEX				•	•													•				
ARCDSEX		•	•														•					
ARCEODEX		•	•			•																
ARCHIVEX		•	•														•					
ARESPREX				•																		
ARPOOLEX				•																		
DARCHIEX		•	•																	•		
DCCMPEX0		•	•			•																
DCCMPEX1		•	•																			
DCCMPEX2		•	•																			
DCCMPEX3		•	•																			
DCCMPEX4		•	•																			
DCCMPEX5		•	•																			
DCCMPEX6		•	•																			
DCCMPEX7		•	•																			
DCCMPEX8		•	•																			
DCCMPEX9		•	•																			
DCCMPEXA		•	•																			
DCCMPEXB		•	•																			
DCCMPEXC		•	•																			

Exit Name	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD	XCOPY
DCCMPEXD		•	•																			
DCCMPEXE		•	•																			
DCCMPEXF		•	•																			
DCDCPEX0		•	•			•	•															
DCDCPEX1		•	•				•															
DCDCPEX2		•	•				•															
DCDCPEX3		•	•				•															
DCDCPEX4		•	•				•															
DCDCPEX5		•	•				•															
DCDCPEX6		•	•				•															
DCDCPEX7		•	•				•															
DCDCPEX8		•	•				•															
DCDCPEX9		•	•				•															
DCDCPEXA		•	•				•															
DCDCPEXB		•	•				•															
DCDCPEXC		•	•				•															
DCDCPEXD		•	•				•															
DCDCPEXE		•	•				•															
DCDCPEXF		•	•				•															
DCDSNDEX		•	•																			
DEFARCEX																						
DERASEEX																				•		
DINXUFEX		•	•	•	•	•	•															
DRESTOEX				•																•		

Exit Name	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD	XCOPY
DSBUSREX																				•	•	
DSCBCVEX				•	•											•	•					
DSCLSCEX	•	•	•			•								•			•					
DSDISPEX		•	•																			
DSNDELEX									•													
DSNDLPEX									•	•												
FMSSAFEX					•																	
FMSSPREX					•																	
GDGCATEX				•	•											•	•					
ICHEKAEX		•	•	•	•	•	•										•	•				
ICHEKPEX		•	•	•	•	•	•										•	•				
IXCTLGEX				•										•								
IXMREVEX									•													
LISTDMEX																				•		
LISTDVEX								•												•		
LISTREEX																				•		
MERDSNEX										•												
MERVOLEX										•												
MIGDISEX															•							
MIGGDGEX															•							
MIGREXIT															•							
RAADDAEX																		•				
RAADDPEX																		•				
RACFDAEX																				•		

Exit Name	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD	XCOPY
RACFDPEX																				•		
RACFEAEX	•	•	•	•	•	•	•	•	•	•							•			•		
RACFEPEX	•	•	•	•	•	•	•	•	•	•							•			•		
RACFSMEX																•						
RACHKAEX	•	•	•	•	•	•	•	•	•	•				•	•	•	•	•		•		
RACHKPEX	•	•	•	•	•	•	•	•	•	•				•	•	•	•	•		•		
RADEFAEX		•	•	•	•	•	•										•	•		•		
RADEFPEX		•	•	•	•	•	•										•	•		•		
RADELAEX									•	•										•		
RADELPEX									•	•										•		
RECPRIEX					•																	
RESAFTEX				•	•		•															
RESPRIEX				•	•		•															
RETAFTEX																						
RETGDGEX																						
RETPRIEX																						
RLSEAFEX															•							
RPTDSBEX	•																				•	
RPTDSNEX	•																					
RPTDSUEX	•																					
RPTMVDEX	•																					
SCRTCHEX		•	•	•	•		•									•	•	•		•		
SECURAEX	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	
SECURPEX	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	

Exit Name	REPORTS	ARCHIVE/BACKUP	BACKUPCC	RESTORE	RECOVER	VBACKUP	VRECOVER	LISTDV	IXMAINT	MERGE	REBUILD	IXUPDATE	FILES MGT	RELEASE	SEQ MIGRATE	PDS COMPRESS	MOVE/COPY	SECURITY	TSS	BILLING	PFD	XCOPY
SPACECEX				•	•													•				
SPFHDGEX	•																					
SPFORDEX																						
SPFORSEX																						
SYSOUTEX	•	•	•	•	•		•	•	•	•	•	•		•	•	•	•	•	•	•	•	
TAOPPOEX		•	•			•				•							•					
TAOPPREX		•	•			•				•							•					
TAPCTLEX									•	•	•	•										
TAPEDLEX									•	•												
TMSCTLEX									•	•	•	•										
TPMGMTX									•	•	•	•										
UNLOADEX						•								•								
USERSEQ		•	•	•	•	•	•									•		•				
USERVENQ																	•					
USRCODEX																				•	•	
VCCOPYEX																	•	•				
VCDMIGEX																	•	•				
VSACESEX		•	•														•					
VSALLOEX				•	•		•										•					
VSATTREX				•	•		•										•					
VSMATFEX																	•					
VSMPRIEX																	•					
VSMSPWEX	•	•	•	•	•		•										•					
VSNEWNEX				•	•												•					

Exit Name		
XCPARCEX		REPORTS
		ARCHIVE/BACKUP
XCPDSNEX		BACKUPCC
		RESTORE
		RECOVER
		VBACKUP
		VRECOVER
		LISTD/
		IXMAINT
		MERGE
		REBUILD
		IXUPDATE
		FILES MGT
		RELEASE
		SEQ MIGRATE
		PDS COMPRESS
		MOVE/COPY
		SECURITY
		TSS
		BILLING
		PFD
	•	XCOPY

User Exit Descriptions

ALLOEXIT — Restore/Recover Data Set Allocation

The module named by the sysparm entry is invoked at allocation time during restore or recover processing for non-VSAM data sets. The format-1 DSCB for the data set being restored is passed to the exit to allow the following fields to be updated prior to its allocation:

- CREATE DATE
- EXPIRE DATE
- LAST USED DATE
- LAST MOD DATE
- OPEN COUNT

Use sysparm ALLOEXIT to specify the module name to be invoked for this exit.

Table 5-1. Parameters Passed to ALLOEXIT

Type	Size	Description
INPUT	XL140	Data Set Identifier - Format-1 DSCB for data set to be processed

ALLOCATEX — Data Set Allocation Preprocessing

The module name supplied for this exit will receive control from any SAMS:Disk routine that allocates non-VSAM data sets on DASD. It receives control prior to allocation (SVC 32 or SVC 99).

The user may request that SAMS:Disk reject the allocation request or continue processing. The format-1 DSCB to be allocated is also provided to the exit for examination and changes.

Use sysparm ALLOCATEX to specify the module name to be invoked for this exit.

Table 5-2. Parameters Passed to ALLOCATEX

Type	Size	Description		
IN/OUT	XL256	DATA SET IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
IN/OUT	XL140	VOLUME IDENTIFIER - Format-4 DSCB for data set. VOLUME SERIAL in positions 1 - 6.		
IN/OUT	XL4	ALLOCATION AMOUNTS - This parameter contains 2 values indicating how much space to allocate. HW - Number of tracks to be allocated HW - Number of directory blocks (if PDS).		
INPUT	XL2	DEVICE TYPE - Device type as documented for UCBTYP field in UCB control block		
OUTPUT	CL100	If user to reject request by setting return code to 2, optionally a message may be placed here. It will be printed by SAMS:Disk. If this field is spaces, no message will be printed.		
OUTPUT	HW	Specify return code:	H'1'	Continue Normal Processing
			H'2'	Reject Allocation Request

ALOPOSEX — Data Set Allocation Post-Processing

The module name supplied for this exit will receive control from any SAMS:Disk routine that allocates non-VSAM data sets on DASD. It receives control after SVC99 allocation.

This exit allows the user to retain the format-1 DSCB data required by their applications in the 13-byte System Code field and the 2-byte Reserved field at offset +103. These are the only two fields where data may be altered and then moved to the actual DSCB. Specify the correct return code to activate update processing.

Use sysparm ALOPOSEX to specify the module name to be invoked for this exit.

Table 5-3. Parameters Passed to ALOPOSEX

Type	Size	Description
IN/OUT	XL256	SOURCE data set IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.
INPUT	XL140	SOURCE VOLUME IDENTIFIER - Format-4 DSCB for data set. VOLUME SERIAL in positions 1 - 6.
IN/OUT	XL256	TARGET data set IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.
INPUT	XL140	TARGET VOLUME IDENTIFIER - Format-4 DSCB for data set. VOLUME SERIAL in positions 1 - 6.
INPUT	XL2	DEVICE TYPE - Device type as documented for UCBTYP field in UCB control block
OUTPUT	HW	RETURN CODE - Specify return code as indicated to have both the System Code field and the Reserved field of the format-1 DSCB updated with user data.
		H'1' Perform Update Processing
		OTHER Continue Normal Processing

ARCDSNEX — Archive Data Set Name Generation Exit

The module name supplied by this exit is invoked at archive data set name generation time. This exit allows you to alter the SAMS:Disk archive data set name to meet special user requirements.

When this user exit is invoked, a unique data set name is created and passed as input. You must make sure that the data set name created by the exit is unique (even across shared operating systems), and that the name selects valid archive pools.

Use sysparm ARCDNEX to specify the module name to be invoked for this exit.

Table 5-4. Parameters Passed to ARCDNEX

Type	Size	Description
IN/OUT	CL44	UNIQUE ARCHIVE DATA SET NAME
INPUT	CL8	JOBNAME
INPUT	CL4	SYSTEM ID as identified to SMF

ARCEODEX — Archive Tape End Of File (CLOSE) Exit

The module name supplied for this exit will receive control after each archive tape is closed. It is used for informational purposes only.

The user is passed the DDNAME in order to determine if it is a close for the primary or the copy tape, and the JFCB in order to see the actual tape volumes used. The data set name and volume serials in the JFCB may be useful for interfacing to tape management systems that read the JFCB at job termination time only, rather than hooking directly into tape open and close routines. It may also be useful for generating tape labels, or for detecting when a multiple volume archive tape was created.

Use sysparm ARCEODEX to specify the module name to be invoked for this exit.

Table 5-5. Parameters Passed to ARCEODEX

Type	Size	Description		
INPUT	H	OPTION - Type of CLOSE call	H'1'	Regular tape close
			H'2'	Final tape close - job is ending
INPUT	CL8	DDNAME	CL8'ARCHIVE0'	Close for primary tape
			CL8'ARCHIVEC'	Close for copy tape
INPUT	XL176	JFCB - The JFCB for the named DDNAME after the tape close has been done		
INPUT	XL110	ARCHVOLS - Image of the ARCHVOLS record as it will be stored in the ARCHVOLS subfile		

ARCHIVEX — Process Archive Request

The module name supplied for this exit will receive control from the SAMS:Disk archive function after it has retrieved the format-1 DSCB and verified the data set exists and is OK to process. The user may request that SAMS:Disk reject the data set for processing or continue on to process the archive request.

Use sysparm ARCHIVEX to specify the module name to be invoked for this exit.

Table 5-6. Parameters Passed to ARCHIVEX

Type	Size	Description			
INPUT	CL8	COMMAND IDENTIFIER - CL8'ARCHIVE'			
INPUT	CL1	ENVIRONMENT	'B'	BATCH	
			'T'	TSS	
OUTPUT	HW	Specify return code	H'1'	Continue Normal Processing	
			H'2'	Terminate Processing For Data Set	
INPUT	XL256	DATA SET IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.			
INPUT	CL44	NEW NAME - If not spaces, NEWNAME for data set			
INPUT	XL140	VOLUME IDENTIFIER - Format-4 DSCB. VOLUME SERIAL in positions 1 - 6.			
IN/OUT	XL3	EXPIRATION DATE - XL3'yydddd' (BINARY)			
IN/OUT	CL4	DISPOSITION INDICATORS	CL1	S	SCRATCH
				N	NOSCRATCH
			CL1	U	UNCAT
				N	NOCAT
				R	RECAT
			CL1	E	Explicit ARCHIVE
			CL1	Y	SIMULATE
				NOT Y	LIVE
INPUT	HW	MEMBER COUNT - If MEMBERS specified			
INPUT	nXL8	MEMBER LIST - List of PDS member names			
INPUT	CL8	Password from PASSWORD command parameter			
INPUT	CL1	RESET parameter	Y	YES	
			N	NO	

ARESPREX — Auto-Restore Screening Exit

The module name supplied for this exit will receive control during the auto-restore function of SAMS:Disk. It is invoked by the auto-restore system task after SAMS:Disk has verified that an index record exists for the data set being restored. It is invoked prior to allocation of the archive device, so if this exit rejects the restore request, no device will be allocated.

The purpose of this exit is to let you screen the auto-restore request and, if desired, reject it. You may also put a logging facility in this exit to track the number and type of auto-restores being processed, or check the device type in the ARCHVOLS record to see if the data set was archived to tape or disk. The only parameter that may be modified by the exit is the result code. Although this exit is invoked by a system task, it is not required that it be reentrant. Take care in coding the exit to ensure that it will not cause an abend. Although the system task will recover and post the waiting task, the restore will not be processed. Your exit should issue an ESTAE macro to set up a recovery environment of its own.

Use sysparm ARESPREX to specify the module name to be invoked for this exit.

Table 5-7. Parameters Passed to ARESPREX

Type	Size	Description		
INPUT	CL44	DATA SET NAME		
INPUT	CL6	VOLUME from which data set was archived		
INPUT	CL6	PSUEDO VOLUME (if the catalog management hook is installed) or TARGET VOLUME (if the s213 hook is installed)		
INPUT	XL256	DSNINDEX RECORD		
INPUT	HW	ADDRESS SPACE ID OF ORIGINATING TASK		
INPUT	CL8	JOB NAME OR TSS ID OF ORIGINATING TASK		
OUTPUT	HW	Specify return code	H'0'	Continue Normal Processing
			H'1'	Defer (queue) the Request
			H'8'	Terminate Processing For Request
INPUT	XL110	ARCHVOLS RECORD		
INPUT	XL167	Auto-restore control block (ARCB) as described by the ARCB macro in the SAMS:Disk macro library		

ARPOOLEX — Auto-Restore DASD Pool Exit

The module name supplied for this exit will receive control during the auto-restore function of SAMS:Disk. It is called only when the auto-restore is driven from the SAMS:Disk catalog management exit, and not the S213 exit. This exit allows the user to inspect the DASD pool that SAMS:Disk has assigned for the restore (if any). If the pool name is blanks on input to the exit, either no pools were defined, or none of the pool definitions matched either the data set name or source volume of the data set.

The exit can either modify the pool name passed on input, or it can pass back an explicit volume to which the data set should be restored. If an explicit volume is returned, the pool name will be ignored. The volume on input will always be blank. If a pool name is returned, it must be a valid pool name defined in the DASDPOOL member of your parmlib data set.

For more information on the use of DASD pools in the auto-restore feature, review the topic *"Setting Up DASD Pools for Auto-Restore"* on page 52 of the *Installation Guide*.

Although this exit is invoked by a system task, the module need not be re-entrant. However, care should be taken in coding the exit so that it will not cause an abnormal termination.

Use sysparm ARPOOLEX to specify the module name to be invoked for this exit.

Table 5-8. Parameters Passed to ARPOOLEX

Type	Size	Description
INPUT	CL44	DATA SET NAME
INPUT	CL6	VOLUME DATA SET WAS ARCHIVED FROM
INPUT	XL256	DSN ARCHIVE INDEX RECORD
INPUT	XL110	DSN ARCHIVE VOLUME RECORD
IN/OUT	CL8	POOL NAME (must be valid dasdpool)
OUTPUT	CL6	EXPLICIT VOLUME TO RESTORE TO

DARCHIEX — Process Deferred Archive Request

The module name supplied for this exit will receive control from the SAMS:Disk DARCHIVE function after it has retrieved the DARCHIVE COMMAND and before the command is placed on the DARCHIVE request queue.

The user may request that SAMS:Disk reject the data set for processing or continue on to queue the DARCHIVE request.

Use sysparm DARCHIEX to specify the module name to be invoked for this exit.

Table 5-9. Parameters Passed to DARCHIEX

Type	Size	Description			
INPUT	CL8	COMMAND IDENTIFIER - CL8'DARCHIVE'			
INPUT	CL1	Environment	'B'	BATCH	
			'T'	TSS/PFD	
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing	
			H'2'	Terminate Processing For Data Set	
INPUT	CL44	DATA SET NAME			
INPUT	CL44	NEW NAME - If not spaces, NEWNAME for data set			
INPUT	CL6	VOLUME - What DASD volume contains data set			
IN/OUT	XL3	EXPIRATION DATE - XL3'yy0ddd' (BINARY)			
IN/OUT	HW	RETENTION PERIOD - Number of days to retain DSN			
IN/OUT	CL4	Disposition Indicators	CL1	S	SCRATCH
				N	NOSCRATCH
			CL1	U	UNCAT
				N	NOCAT
				R	RECAT
			CL1	(unused)	
			CL1	Y	SIMULATE
				NOT Y	LIVE
INPUT	HW	MEMBER COUNT - If MEMBERS specified			
INPUT	nXL8	MEMBER LIST - List of PDS member names			
INPUT	CL8	Password from PASSWORD command parameter			
INPUT	CL1	RESET parameter	Y	YES	
			N	NO	

DCCMPEXn — Data Compression

The module name supplied for this exit will be given control to compress archive records. Technique number (n) is specified by sysparm DCCOMPTC. Only technique numbers 8 through F may be specified for user exits.

Use sysparm DCCMPEXn to specify the module name to be invoked for this exit.

Table 5-10. Parameters Passed to DCCMPEXn

Type	Size	Description
INPUT	FW	INPUT RECORD LENGTH - length of the record in parameter 2
INPUT	XL	ARCHIVE RECORD - data record to be compressed
OUTPUT	FW	OUTPUT RECORD LENGTH - place the length of compressed record here
OUTPUT	XL	OUTPUT RECORD BUFFER - A 45,056-byte buffer to receive the compressed record

Note: Your exit must be able to process records with input length of zero.

DCDCPEXn — Data Decompression

The module name supplied for this exit will be given control to decompress archive records. The exit is passed the record and the length of the record. The exit must decompress the record into the buffer being passed, and return the length of the decompressed record.

Use sysparm DCDCPEXn to specify the module name to be invoked for this exit.

Table 5-11. Parameters Passed to DCDCPEXn

Type	Size	Description
INPUT	FW	INPUT RECORD LENGTH - length of compressed record
INPUT	XL	COMPRESSED RECORD
OUTPUT	FW	OUTPUT RECORD LENGTH - place the length of the decompressed record here
OUTPUT	XL	OUTPUT RECORD BUFFER - A 45,056-byte buffer to receive the decompressed record

Note: Your exit must be able to process records with input length of zero.

DCDSNDEX — Screen Data Sets for Data Compression

The module name supplied for this exit is given control before doing data compression for a data set, to determine:

1. should compression be done on this data set
2. what technique to use for compression

Use sysparm DCDSNDEX to specify the module name to be invoked for this exit.

Table 5-12. Parameters Passed to DCDSNDEX

Type	Size	Description		
INPUT	CL44	DATA SET NAME		
IN/OUT	C	TECHNIQUE NUMBER - change this to indicate the data compression technique used (C'0' - C'F')		
IN/OUT	C	DATA COMPRESSION INDICATOR - change this to whether or not data compression is to be done	C'Y'	DO data compression
			C'N'	NO data compression

DEFARCEX — Screen Disposal of Deferred Archive Requests

During dispose processing of deferred (queued) archive requests, the module named for this exit is given control after an entry is retrieved from the queue, but before it has been selected for disposal (just before the data set is processed). The return code indicates if SAMS:Disk should archive the data set or flag this command and retrieve the next one. If the command is flagged as rejected, it will not be reviewed again, and will be dropped from the ARCHCMDS file when the hold period (specified in sysparm DARCKEEP) has expired.

Use sysparm DEFARCEX to specify the module name to be invoked for this exit.

Table 5-13. Parameters Passed to DEFARCEX

Type	Size	Description			
INPUT	CL44	DATA SET NAME			
INPUT	CL6	VOLUME SERIAL NUMBER			
INPUT	CL2	Disposition ACTION	CL1	S	SCRATCH
				N	NOSCRATCH
			CL1	U	UNCATALOG
				N	NOCATALOG
				R	RECATALOG
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing	
			H'2'	Reject Processing For Command	

DERASEEX — Process Erase TSS Deferred Processing Request

The module name supplied for this exit will receive control from the SAMS:Disk DERASE function after it has retrieved the DERASE command and before the specified request is deleted.

The user may request that SAMS:Disk reject the data set for request or continue on to delete the request.

Use sysparm DERASEEX to specify the module name to be invoked for this exit.

Table 5-14. Parameters Passed to DERASEEX

Type	Size	Description		
INPUT	CL8	COMMAND IDENTIFIER - CL8'DERASE'		
INPUT	CL1	Environment	'T'	TSS
			'B'	Batch
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Request
INPUT	CL44	DATA SET NAME - DSN for record to be deleted		
INPUT	CL1	TYPE TO BE DELETED	A	DARCHIVE
			R	DRESTORE

DINXUFEX — Update DSNINDEX User Field

The module name supplied for this exit will receive control from SAMS:Disk before creating or updating the DSNINDEX record in the Files Data Set. It is invoked during processing for the ARCHIVE, BACKUP, VBACKUP, RESTORE, RECOVER and VRECOVER functions. For each data set being processed by one of these functions, this exit is invoked, just before the DSNINDEX record is written to the Files Data Set.

SAMS:Disk allows only one field in the DSNINDEX record to be updated by this exit; the 25-byte DSNUSERF user field. SAMS:Disk ignores any other changed fields passed back in the DSNINDEX record.

Use sysparm DINXUFEX to specify the module name to be invoked for this exit.

Table 5-15. Parameters Passed to DINXUFEX

Type	Size	Description		
IN/OUT	XL256	DSNINDEX RECORD - This is the archive data set index record from the files data set. DSNUSERF in positions 231 - 256 is the only field for which updates will be honored. Turn to page 529 for a complete description of this record.		
INPUT	CL8	SAMS:Disk Function - This is the function processing when the exit is invoked.	Valid Values:	CL8'ARCHIVE '
				CL8'BACKUP '
				CL8'VBACKUP '
				CL8'RESTORE '
				CL8'RECOVER '
				CL8'VRECOVER'

DRESTOEX — Screen Deferred Restore Request

The module name supplied for this exit will receive control from the SAMS:Disk DRESTORE function after it has retrieved a DRESTORE command and before the command is placed on the DRESTORE request queue.

The user may request that SAMS:Disk reject the data set for processing, or continue on to queue the DRESTORE request.

Use sysparm DRESTOEX to specify the module name to be invoked for this exit.

Table 5-16. Parameters Passed to DRESTOEX

Type	Size	Description			
INPUT	CL8	Command Identifier	CL8'DRESTORE'		
INPUT	CL1	Environment	'B'	Batch	
			'T'	TSS/PFD	
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing	
			H'2'	Data Set Processing Terminated	
INPUT	CL44	DATA SET NAME			
INPUT	CL44	NEW NAME - If not spaces, NEWNAME for data set			
IN/OUT	CL6	VOLUME - What DASD volume contains data set			
IN/OUT	XL3	EXPIRATION DATE - XL3'yydddd' (BINARY)			
IN/OUT	HW	Retention Period - Number of days to retain DSN			
IN/OUT	CL4	Disposition Indicators	CL1	S	SCRATCH specified
			CL1	N	NOCAT specified
				R	RECAT specified
			CL1	C	CREATE specified
			XL1	X'01'	ERASE specified
INPUT	HW	MEMBER COUNT - If MEMBERS specified			
INPUT	nXL8	MEMBER LIST - List of PDS member names			
INPUT	CL9	Data from TIMEDATE parameter			
IN/OUT	CL8	DASDPOOL name from POOL parameter			
INPUT	CL8	Password from PASSWORD command parameter			

DSBUSREX — Screen Billing Record Inserts

This module is given control during DASD billing just prior to inserting or updating a data set billing record in the files data set. Its intended use is to allow the user code field to be reset.

Use sysparm DSBUSREX to specify the module name to be invoked for this exit.

Table 5-17. Parameters Passed to DSBUSREX

Type	Size	Description		
INPUT	CL2	Type Of Function	'SI'	Insert record into file
			'SR'	Replace existing record
IN/OUT	XL	Dasd Billing Record - Turn to page 526 for a complete description of the fields in this record		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Ignore This Insert / Replace Request

DSCBCVEX — Format-1 DSCB Conversion Exit Location

The module name supplied for this exit will receive control after a new user data set has been allocated on DASD (for example, during Restore, Move/Copy, and PDS Compress processing) and by the DSCB Update Utility. This exit may update or move values of SAMS:Disk maintained fields kept in the format-1 DSCB.

A sample conversion exit has been provided in member ADSEX016 of the installation library. This sample exit will convert format-1 DSCBs maintained with an Open SVC from a release prior to 8.1 to the Release 8.1 offsets.

By default, the supplied conversion exit will retain the last using job name, move the last modified date, reset the open count, and update the SVC MODE. This will only do conversion if the current SVC mode is 1, 2 or 3 (indicating that it has not be converted before).

Conversion is done using the old default offsets. If you have modified the Open SVC and maintained fields at different offsets, you must modify this exit to reflect your changes. Also, if you wish to maintain the creating job name instead of the last using job name, you must update the exit to reflect this.

You should not begin using this conversion exit until you have the new Open SVC installed. Since the old Open SVC does not know of the new offsets, these fields can become corrupted after they are converted.

Once you have determined that this exit will work as desired, specify sysparm DSCBCVEX with the value of ADSEX016 or whatever you have named your con-

version exit. This value will cause conversion to take place as SAMS:Disk reallocates data sets during its processing.

Table 5-18. Parameters Passed to DSCBCVEX

Type	Size	Description		
IN/OUT	XL140	Data Set Identifier. Format-1 DSCB to be converted. This format-1 DSCB is as it was prior to the current SAMS:Disk processing. You must update this format-1 DSCB to reflect the proper values at the new offsets.		
IN/OUT	XL140	Data Set Identifier. Format-1 DSCB to be converted. This format-1 DSCB also needs to be converted. This format-1 DSCB also contains other updates made by SAMS:Disk function currently processing the data set.		
IN	H	Count Of Fields To Be Changed. This was input to the calling module. It identifies the number of other fields to be updated at the same time as the fields being converted.		
IN	nCL8	Field Name List. This is a list of the field names to be updated at the same time as the fields being converted by this exit. The values of these fields are already in the format-1 DSCB identified in parameter 2.		
OUT	H	Return Code	H'1'	Continue Normal Processing
			Other	Indicates Errors

DSCLSCEX — DSCL Screening Exit

The module name supplied for this exit will receive control when processing any DSCL command. Control is passed after the screening tests have been applied, and directly prior to processing the DSCL action command. The screening tests are based on the DSCL selection criteria, after the volume and data set name filtering has been completed. The user may request that SAMS:Disk reject or process the request, regardless of what value was found in the result code parameter.

When the DS1CPOIT bit in the VTOC is set to 1 (on), the user exit DSCLSCEX allows the action commands ARCHIVE, MOVE, EXPIRE, RELEASE, or DELETE to take place only if an installation determined *number of days* has passed since the data set was last referenced. The default value of 0 allows the data set to be moved regardless of the value in the last used date.

Use system parameter DSCLSCEX to specify the module name to be invoked for this exit.

Table 5-19. Parameters Passed to DSCLSCEX

Type	Size	Description		
IN/OUT	HW	The value on input informs the exit what SAMS:Disk action will take place if unmodified by the exit. The value on output (if changed) instructs SAMS:Disk to take the new action.	H'0'	Continue Normal Processing
			H'8'	Terminate Processing For Request
INPUT	XL256	Data Set Identifier. Format-1 DSCB and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL140	Volume Identifier. Format-4 DSCB for data set. Volume serial is in position 1 to 6. If the data set to be processed is VSAM, only the volume serial will be passed.		
INPUT	CL44	Catalog Name - If VSAM, the catalog name in which the component is defined (if catalog scan)		

Type	Size	Description		
INPUT	CL8	Function Code	'ARCHIVE'	archive the data set
			'BACKUP'	backup the data set
			'COPY'	copy the data set
			'DELETE'	delete the data set
			'EXPIRE'	expire the data set
			'MOVE'	move the data set
			'RELEASE'	release space from the data set
			'REPORT'	report on the data set
			'VBACKUP'	backup a volume
INPUT	CL8	Disposition Value of DISP= DSCL ARCHIVE action command parameter.	'DELETE'	delete data set
			'KEEP'	do not scratch or recatalog set
			'RECAT'	recatalog data set to the SAMS:Disk pseudo volser
OUTPUT	F	This optional parameter indicates the number of days since last referenced for the data set to be eligible for processing. The default value is '0'.		

DSDISPEX — Post Data Set Disposition Action

A post-disposition exit, this exit is given control during backup and archive processing, after dispose processing is complete.

Use sysparm DSDISPEX to specify the module name to be invoked for this exit.

Table 5-20. Parameters Passed to DSDISPEX

Type	Size	Description		
INPUT	XL256	DATA SET IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	CL4	DISPOSITION FIELD - Disposition field in the TASC format, found in RETVOLSPEC table, indicates the action SAMS:Disk performed on the data set.		
INPUT	XL140	VOLUME IDENTIFIER - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.		
INPUT	XL1	SIMULATION indicator	X'01'	If in SIMULATE mode
			X'00'	otherwise it will contain this

DSNDELEX — Screen Index Maintenance Deletes (Prior Exit)

The module name supplied for this exit will receive control from the SAMS:Disk index maintenance functions VOLDELETE, DSNDELETE and PURGE prior to making a decision to delete an index record for an archived data set from the DSNINDEX. The exit routine will receive control for every index record scanned, prior to any SAMS:Disk decisions to determine if it should be kept or deleted.

The user may request that SAMS:Disk continue to decide what to do with the record, to delete it, or to keep it in the DSNINDEX.

The Common File Handler Interface (described on page 30 in this manual), should not be used from this exit to update a record. The update you perform can be lost when you return control and the original function rewrites the record as initially passed to the exit.

Use sysparm DSNDELEX to specify the module name to be invoked for this exit.

Table 5-21. Parameters Passed to DSNDELEX

Type	Size	Description		
INPUT	XL256	DSNINDEX RECORD - This is the archive data set index record from the files data set. DSNAME is in positions 1 - 44. Turn to page 529 for a complete description of this record.		
INPUT	XL1	SIMULATION indicator	X'01'	If in SIMULATE mode
			X'00'	otherwise it will contain this
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Keep Record, Do Not Delete It
			H'3'	Delete Record

DSNDLPEX — Screen Index Maintenance Deletes (After Exit)

The module name supplied for this exit will receive control from the SAMS:Disk index maintenance functions VOLDELETE, DSNDELETE and PURGE, the SAMS:Disk index update function DELETE, the PFD function LIST DSNINDEX AND CHANGE EXPIRATION DATE, and the SAMS:Disk function MERGE after SAMS:Disk has determined whether an index record is to be deleted, but prior to actually deleting it from the DSNINDEX. The user may request that SAMS:Disk continue to take the indicated action against the data set or override the decision made by SAMS:Disk. The action to be taken by SAMS:Disk is indicated in the third parameter RESULT.

The Common File Handler Interface (described on page 30 in this manual), should not be used from this exit to update a record. The update you perform can be lost when you return control and the original function rewrites the record as initially passed to the exit.

Use sysparm DSNDLPEX to specify the module name to be invoked for this exit.

Table 5-22. Parameters Passed to DSNDLPEX

Type	Size	Description			
INPUT	XL256	DSNINDEX Record - This is the archive data set index record from the files data set. DSNAME is in positions 1 - 44. Turn to page 529 for a complete description of this record.			
INPUT	XL1	Simulation indicator	X'01'	If in simulate mode	
			X'00'	otherwise it will contain this	
IN/OUT	HW	RESULT - SAMS:Disk Input is H'1' or less indicating delete the DSNINDEX record, and H'2' to keep the record. If the user wishes SAMS:Disk to reject the delete request and keep the DSNINDEX record, place a value of 2 in this parameter and return to SAMS:Disk; otherwise place a H'1' in this parameter. If the Common File Handler Interface (described on page 30) is used in this exit, use the return codes that specify repositioning required.		H'0'	Delete the DSNINDEX record, and reposition it
				H'1'	Delete the DSNINDEX record
				H'2'	Keep the DSNINDEX record
				H'3'	Keep and reposition the DSNINDEX record

FMSSAFEX — FMS After Selection Exit

The module name supplied for this exit receives control after standard FMS selection processing is complete for the DSNINDEX record. This exit is passed a copy of the DSNINDEX record selected for recovery, and a Y/N flag byte.

The initial setting of the flag is either C'Y' or C'N', indicating whether or not DSCL processing selected the data set. By setting this flag to C'Y', the data set will be selected regardless of any further DSCL statements. This flag can also be set to C'N', forcing a bypass of the data set regardless of any further DSCL statements.

Use sysparm FMSSAFEX to specify the module name to be invoked for this exit.

Table 5-23. Parameters Passed to FMSSAFEX

Type	Size	Description		
INPUT	CL256	Copy of the DSNINDEX record		
IN/OUT	CL1	Flag Byte Return	C'Y'	Select the data set
			C'N'	Bypass the data set

FMSSPREX — FMS Prior Selection Exit

The module name supplied for this exit receives control prior to FMS performing any selection processing for the DSNINDEX record. This exit is passed a copy of the DSNINDEX record selected for recovery, and a Y/N flag byte.

The initial setting of the flag is a hex value of zero (X'00'), indicating normal selection processing will continue. By setting this flag to C'Y', the data set will be selected regardless of any further DSCL statements. This flag can also be set to C'N', forcing a bypass of the data set regardless of any further DSCL statements.

Use sysparm FMSSPREX to specify the module name to be invoked for this exit.

Table 5-24. Parameters Passed to FMSSPREX

Type	Size	Description		
INPUT	CL256	Copy of the DSNINDEX record		
IN/OUT	CL1	Flag Byte Return	C'Y'	Select the data set
			C'N'	Bypass the data set

GDGCATEX — GDG Data Set Catalog Action

The module name supplied for this exit will receive control during functions RESTORE, RECOVER, MOVE, COPY, PDS COMPRESS, and IXCATLG prior to any catalog action for GDG data sets. A sample coding of this exit can be found in member GDGCATEX in the INSTALL library.

Default processing will not force a GDG catalog entry out of a full GDG sphere, but instead will leave the processed GDS in a deferred or uncataloged status. This exit gives the user the ability to either force the catalog action, or allow normal processing to take its course.

The GDG Status parameter, described in the following table, indicates the catalog status of the source data set (i.e., BACKUP, MOVE, or COPY).

Use sysparm GDGCATEX to specify the module name to be invoked for this exit.

Table 5-25. Parameters Passed to GDGCATEX

Type	Size	Description		
INPUT	CL44	Data set name or NEWNAME (if specified)		
INPUT	CL1	GDG Status	H	Active
			N	Deferred
			M	Rolled Off
OUTPUT	CL1	Catalog Action requested	C	Unconditionally catalog (or Roll-in) the GDS
			, ,	Continue with normal processing
OUTPUT	H	Results:	0	Process according to Catalog Action setting
			1+	Bypass any catalog action. M3729 will be issued.

ICHEKAEX — RACF ICHEINTY Processing Post-ICHEINTY Exit

The module name supplied for this exit will receive control from all functions of SAMS:Disk that need to determine whether a data set has a discrete profile or a generic profile.

SAMS:Disk issues the Fujitsu ICHEINTY macro to determine if a data set has a discrete and/or a generic profile. This exit is invoked after issuing this macro.

The user is given an opportunity in this exit to look at the data set name and volume after issuing the ICHEINTY macro. The user is also given an opportunity to review the return code passed back from the macro. The return codes issued by the ICHEINTY macro are documented in the Fujitsu SPL RACF manual.

Use sysparm ICHEKAEX to specify the module name to be invoked for this exit.

Table 5-26. Parameters Passed to ICHEKAEX

Type	Size	Description	
INPUT	XL1	D	SAMS:Disk checks to see if the data set has a discrete profile.
		G	SAMS:Disk checks to see if the data set has a generic profile.
INPUT	XL44	DATA SET NAME TO BE CHECKED	
INPUT	XL6	VOLSER OF DATA SET TO BE CHECKED. If parameter 1 is G, VOLSER is ignored.	
INPUT	HW	RETURN CODE - returned from the ICHEINTY macro	

ICHEKPEX — RACF ICHEINTY Processing Pre-ICHEINTY Exit

The module name supplied for this exit will receive control from all functions of SAMS:Disk that need to determine whether a data set has a discrete profile or a generic profile.

SAMS:Disk issues the Fujitsu ICHEINTY macro to determine if a data set has a discrete and/or a generic profile. This exit is invoked prior to issuing this macro.

The user may modify the data set name or the VOLSER. The user may also prevent the issuing of the macro by changing the return code to a value greater than zero. If the value is changed to greater than zero, this is equivalent to saying to SAMS:Disk that the data set does not have either a discrete or a generic profile, depending on what type of profile is being looked for.

Use sysparm ICHEKPEX to specify the module name to be invoked for this exit.

Table 5-27. Parameters Passed to ICHEKPEX

Type	Size	Description	
INPUT	XL1	D	SAMS:Disk determines if the data set has a discrete profile.
		G	SAMS:Disk determines if the data set has a generic profile.
IN/OUT	XL44	DATA SET NAME TO BE CHECKED	
IN/OUT	XL6	VOLSER OF DATA SET TO BE CHECKED. If parameter 1 is G, VOLSER is ignored.	
OUTPUT	HW	RETURN CODE. Set to a value greater than zero if you wish the ICHEINTY macro to not be issued.	

IXCTLGEX — User Exit for IXCATLG Utility

This module is invoked after all selection testing has been performed by the IXCATLG procedure. It can be used either to monitor those data sets selected by the maintenance program or to change the selection flag to either include or exclude the data set.

When this exit is invoked, the selection flag has already been set to the value SAMS:Disk has determined is correct. If the value is Y, the data set will be cataloged to the SAMS:Disk pseudo-volume as specified by sysparm RECATVOL, to enable it as a candidate to be auto-restored. If the value is N, the data set will not be processed. This exit can change the selection flag to either a Y or an N, regardless of which input value was passed to the exit.

Use sysparm IXCTLGEX to specify the module name to be invoked for this exit. (A sample exit is provided in the SAMS:Disk installation library as member IXCTLGEX.)

Table 5-28. Parameters Passed to IXCTLGEX

Type	Size	Description		
INPUT	XL256	DSNINDEX record		
IN/OUT	CL1	Selection flag	Y	data set has been selected for processing
			N	data set has been excluded from processing

IXMREVEX — Screen Index Maintenance Deletes (Post Exit)

Use sysparm *IXMREVEX* described on page 155 to specify the module name to be invoked for this exit. This exit receives control from the DSNDELETE, PURGE, and VOLDELETE commands after SAMS:Disk has determined what action to take on the DSNINDEX record, but prior to actually taking the action. The action to be taken by SAMS:Disk is indicated in the third parameter RETURN CODE. The user may request that SAMS:Disk continue to take the indicated action against the data set, or you can override the action entirely.

Caution: The *Common File Handler Interface* described beginning on page 30 in this manual, should **NOT** be used from this exit to update a record. If you use this interface to update a record, the update you perform can be lost when returning control and the original function rewrites the record initially passed to the exit.

The DSNDLPEX user exit must not be specified at the same time as this exit. If both are specified, the job will abend with a U100 accompanied by an appropriate message.

Table 5-29. Parameters Passed to IXMREVEX

Type	Size	Description		
INPUT	XL256	DSNINDEX record		
INPUT	XL1	Simulation Flag	X'01'	Simulation mode
			X'00'	Live mode
IN/OUT	HW	Return Code	H'0'	Record will be kept — Input
			H'1'	Record will be deleted — Input
			H'2'	Record will be kept — Input
			H'3'	Record will be updated — Input
			H'1'	Delete DSNINDEX record — Output
			H'2'	Keep DSNINDEX record — Output
		Note: Do not set this field unless you want to alter its disposition. An output return code of other than its original input return code or 1 or 2 will result in an abend.		
INPUT	CL16	WHYFLAGS		
INPUT	XL256	Catalog locate work area		
INPUT	XL212	Management Class		

Additional Notes:

1. If a Catalog Locate was performed on the data set, then its workarea is contained in parm 5. Offset 6 contains the data set's cataloged volser.
2. If the data set has a Management Class, then the SAMS:Disk mapping is contained in parm 6 as mapped out by the SMSMC mapping macro.

LISTDMEX — Screen TSS LISTDMS Command

The module name supplied for this exit will receive control from the SAMS:Disk LISTDMS function after it has retrieved the LISTDMS command and before it is processed.

The user may request that SAMS:Disk reject the request or continue processing it.

(Turn to page [249](#) for related information.)

Use sysparm LISTDMEX to specify the module name to be invoked for this exit.

Table 5-30. Parameters Passed to LISTDMEX

Type	Size	Description			
INPUT	CL8	Command Identifier - CL8'LISTDMS'			
INPUT	CL1	Environment	'T'	TSS (always)	
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing	
			H'2'	Terminate Processing For Request	
INPUT	CL44	Data Set Name - DSN for record to be listed			
INPUT	CL3	Data To Be Listed	N	DON'T LIST	
			Y	CL1	List Data Set Status
				CL1	List Data Set Allocation Data
				CL1	List Data Set Archive Data

LISTDVEX — List Archive Index (LISTD/LISTV) Exit

The module name supplied for this exit will receive control after each ARCHVOLS or DSNINDEX entry has been selected for printing. The exit, if used, is also active under the TSS LISTDMS function.

The user is passed the file name (ARCHVOLS or DSNINDEX), the index record itself, and a one-character switch that can be set to suppress the printing of the record.

Use sysparm LISTDVEX to specify the module name to be invoked for this exit.

Table 5-31. Parameters Passed to LISTDVEX

Type	Size	Description			
INPUT	CL8	Filename	CL8'ARCHVOLS'		
			CL8'DSNINDEX'		
INPUT	XLnnn	Index Record For Named File - Turn to page 525 for field lengths within the ARCHVOLS subfile, or page 529 for field lengths within the DSNINDEX subfie.			
IN/OUT	CL1	Print Switch	CL1	'Y'	Display the index entry
			CL1	'N'	Do NOT display the entry

LISTREEX — Screen TSS LISTREQ Command

The module name supplied for this exit will receive control from the SAMS:Disk LISTREQ function after it has retrieved the LISTREQ command and before it is processed.

The user may request that SAMS:Disk reject the request or continue processing it.

Use sysparm LISTREEX to specify the module name to be invoked for this exit.

Table 5-32. Parameters Passed to LISTREEX

Type	Size	Description		
INPUT	CL8	Command Identifier - CL8'LISTREQ'		
INPUT	CL1	Environment	'T'	TSS (always)
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Request
INPUT	CL44	Data Set Name - DSN for record to be listed		
INPUT	CL1	Type To Be Listed	A	DARCHIVE
			R	DRESTORE

MERDSNEX — Data Set Being Deleted Through Merge Processing

The module name supplied for this exit will receive control from the SAMS:Disk MERGE function after an archive DSNINDEX record has been deleted from the files data set.

No return code is accepted by this exit. The MERGE process cannot be altered.

Use sysparm MERDSNEX to specify the module name to be invoked for this exit.

Table 5-33. Parameters Passed to MERDSNEX

Type	Size	Description		
INPUT	XL256	DSNINDEX RECORD - This is the archive data set index record from the files data set. Data set name is in positions 1 to 44. Turn to page 529 for a complete description of this record.		
INPUT	XL1	Simulation Indicator	X'01'	If in SIMULATE mode
			X'00'	otherwise it will contain this

MERVOLEX — Archive Volume Being Deleted By Merge Processing

The module name supplied for this exit will receive control from the SAMS:Disk MERGE function after an ARCHVOLS record has been deleted from the files data set. (If a multivolume ARCHVOLS entry was deleted, only the first ARCHVOLS record in the chain is passed to this exit. Use exit TAPEDLEX (documented on page 289 in this manual) instead of this one if you need to get the list of all ARCHVOLS that were in the chain.)

No return code is accepted by this exit. The MERGE process cannot be altered.

Use sysparm MERVOLEX to specify the module name to be invoked for this exit.

Table 5-34. Parameters Passed to MERVOLEX

Type	Size	Description		
INPUT	XL110	ARCHVOLS RECORD - This is the archive VOLUME index record from the files data set. Volume is in positions 1 - 6. Turn to page 525 for a complete description of this record.		
INPUT	XL1	Simulation indicator	X'01'	If in SIMULATE mode
			X'00'	otherwise it will contain this

MIGDISEX — Screen Sequential Migrate Disposition Request

The module name supplied for this exit will receive control from sequential migrate after it has copied a data set to tape, but before it has scratched/recataloged it. The exit may examine disposition fields and optionally reject the disposition request.

Use sysparm MIGDISEX to specify the module name to be invoked for this exit.

Table 5-35. Parameters Passed to MIGDISEX

Type	Size	Description		
INPUT	XL1	Simulate Mode Indicator - X'01' implies a SIMULATE execution of SAMS:Disk. X'00' implies a LIVE run.		
INPUT	XL147	Disposition Record - This record contains information about the data set copied to tape. Its format is as follows:		
		+00	CL30	1 to 5 OUTPUT TAPE VOLSERS
		+30	CL44	DATA SET NAME
		+74	XI1	DS1DSIND field from the FMT 1 (contains RACF & PASSWORD bits)
		+75	XI3	(RESERVED)
		+78	HW	NUMBER TAPE VOLUMES USED
		+80	CL30	VOLSERS OF COPY TAPES
		+110	HW	DATA SET SEQUENCE NUMBER ON TAPE
		+112	XL3	(RESERVED)
		+115	XL3	EXPIRATION DATE XL3'YYDDDD'
		+118	CL8	(RESERVED)
		+126	CL6	VOLUME MIGRATED FROM
		+132	XL2	DATA SET LRECL
		+134	XL2	DATA SET BLKSIZE
		+136	XL2	TRACKS ALLOCATED
		+138	XL2	DEVICE TYPE OF DASD
		+140	XL1	DATA SET RECFM
		+141	XL1	DATA SET KEY LENGTH
		+142	XL1	DATA SET OPTCD
		+143	CL4	REASON DATA SET MIGRATED
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Do Not Take Disposition For Data Set

MIGGDGEX — Return GDG Limit to Sequential Migrate for a DSN

This module is given control after sequential data set migration has failed to obtain a generation limit from the GDGDASDG table for a GDG data set.

The user may supply the GDG limit for the data set.

Use sysparm MIGGDGEX to specify the module name to be invoked for this exit.

Table 5-36. Parameters Passed to MIGGDGEX

Type	Size	Description		
INPUT	CL44	DATA SET NAME		
OUTPUT	HW	GDG LIMIT COUNT to use for the data set		
OUTPUT	HW	Specify Return Code	H'1'	Generation Limit Returned
			H'2'	Use Default Generation Limit

MIGREXIT — Sequential Migration Disposition Screening

This module is invoked after a data set has been selected for sequential data set migration, but before a disposition control record has been built.

The user may reject migration of the data set or alter DSCB information for a data set to be migrated.

Use sysparm MIGREXIT to specify the module name to be invoked for this exit.

Table 5-37. Parameters Passed to MIGREXIT

Type	Size	Description		
IN/OUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area. Only the expiration date may be modified; all other changes will be ignored.		
INPUT	L140	Volume Identifier - Format-4 DSCB for data set. VOLSER is in positions 1 to 6.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Data Set

RAADDAEX — RACF RACDEF-ADDVOL Post Processing Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for adding volumes to RACF profiles. It receives control after the RACDEF-ADDVOL macro has been issued.

The exit may inspect parameters passed to RACF and change the return code returned from it.

Use sysparm RAADDAEX to specify the module name to be invoked for this exit.

Table 5-38. Parameters Passed to RAADDAEX

Type	Size	Description		
INPUT	CL44	Profile Name That Was Modified		
INPUT	CL6	OLD VOLUME - Existing volume in the profile data after which the new volume was added.		
INPUT	CL6	NEW VOLUME - New volume serial associated with the RACF profile.		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
IN/OUT	HW	The RETURN CODE from the RACDEF (ADDVOL) is placed in this parameter. It may be changed by the exit. Use one of the following valid values:		
		H'0'	Volume Added	
		H'4'	Resource Previously Defined	
		H'8'	Failed By Racf Exit	
		H'16'	Critical Error	

RAADDPEX — RACF RACDEF-ADDVOL Processing Pre-RACHECK Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for adding volumes to RACF profiles. It receives control before the RACDEF-ADDVOL macro is issued.

The exit may change RACDEF-ADDVOL parameters, or cause the RACDEF-ADDVOL to be bypassed and return its own return code.

Use sysparm RAADDPEX to specify the module name to be invoked for this exit.

Table 5-39. Parameters Passed to RAADDPEX

Type	Size	Description		
INPUT	CL44	Profile Name To Be Modified		
INPUT	CL6	OLD VOLUME - Existing volume in the profile data after which the new volume is to be added.		
INPUT	CL6	NEW VOLUME - New volume serial to be associated with the RACF profile.		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
OUTPUT	HW	If your exit performs the RACDEF ADDVOL, set the appropriate return code, as indicated below. If SAMS:Disk is to process the RACDEF ADDVOL function, do not set this return code.		
		H'0'	Volume Added	
		H'4'	Resource Previously Defined	
		H'8'	Failed By Racf Exit	
		H'16'	Critical Error	

RACFDAEX — Decode SAMS:Disk RACF Encoded Profile Name (Post-decode)

The module name supplied for this exit will receive control from the SAMS:Disk RACF processing when a request has been made to decode a SAMS:Disk RACF encoded profile name. Control is passed to the user after SAMS:Disk has decoded the profile name.

The user may elect to change one of the fields from the decoded name or leave them as they are.

Use sysparm RACFDAEX to specify the module name to be invoked for this exit.

Table 5-40. Parameters Passed to RACFDAEX

Type	Size	Description		
INPUT	XL44	SAMS:Disk RACF ENCODED PROFILE NAME — Format of this name is described on page 402		
IN/OUT	XL3	ARCHIVE DATE - Date from decoded profile name. XL3'YYDDDD' (BINARY).		
IN/OUT	XL2	ARCHIVE TIME - Time from decoded profile name. XL2'HHMM' (BINARY UNSIGNED).		
IN/OUT	CL44	DSNAME - Data set name from decoded profile.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Processing, Successful
			H'2'	Error, Reject Decode Request

RACFDPEX — Decode SAMS:Disk RACF Encoded Profile Name (Pre-decode)

The module name supplied for this exit will receive control from the SAMS:Disk RACF processing when a request has been made to decode a SAMS:Disk RACF encoded profile name. Control is passed to the user prior to decoding the profile name.

The user may elect to decode the name, reject the decode request, or indicate to continue processing without change. This exit allows users to implement their encoding technique for SAMS:Disk RACF profile names.

Use sysparm RACFDPEX to specify the module name to be invoked for this exit.

Table 5-41. Parameters Passed to RACFDPEX

Type	Size	Description		
INPUT	XL44	SAMS:Disk RACF ENCODED PROFILE NAME — Format of this name is described on page 402		
OUTPUT	XL3	ARCHIVE DATE - Date from decoded profile name. XL3'YYDDDD' (BINARY).		
OUTPUT	XL2	ARCHIVE TIME - Time from decoded profile name. XL2'HHMM' (BINARY UNSIGNED).		
OUTPUT	CL44	DSNAME - Data set name from decoded profile.		
OUTPUT	HW	Specify Return Code	H'1'	User Exit Decoded The Encoded Profile Name
			H'2'	Continue With SAMS:Disk Decode Of Profile Name
			H'3'	Error, Reject Decode Request

RACFEAEX — Encode SAMS:Disk RACF Profile Name (Post-en-code)

The module name supplied for this exit will receive control from SAMS:Disk RACF processing when a request has been made to encode a SAMS:Disk data set name into a SAMS:Disk RACF profile encoded name. Control is passed to the exit after the name has been encoded.

The user may change the encoded name or leave it as it is.

Use sysparm RACFEAEX to specify the module name to be invoked for this exit.

Table 5-42. Parameters Passed to RACFEAEX

Type	Size	Description		
INPUT	CL44	DSNAME - Encoded data set name.		
INPUT	XL3	ARCHIVE DATE - Date used in encoding the name. XL3'YYDDDD' (BINARY).		
INPUT	XL2	ARCHIVE TIME - Time used in encoding the name. XL2'HHMM' (BINARY UNSIGNED).		
IN/OUT	xL44	SAMS:Disk RACF ENCODED PROFILE NAME		
OUTPUT	HW	Specify Return Code	H'1'	Continue Processing, Successful
			H'2'	Error, Reject Encode Request

RACFEPEX — Encode SAMS:Disk RACF Profile Name (Pre-encode)

The module name supplied for this exit will receive control from SAMS:Disk RACF processing when a request has been made to encode a SAMS:Disk data set name into a SAMS:Disk RACF profile encoded name. Control is passed to the exit prior to encoding the profile name.

The user may elect to encode the name, reject the encode request, or indicate to continue processing without change. This exit allows users to implement their own encoding technique for SAMS:Disk RACF profile names.

Use sysparm RACFEPEX to specify the module name to be invoked for this exit.

Table 5-43. Parameters Passed to RACFEPEX

Type	Size	Description		
INPUT	CL44	DSNAME - Data set name to be encoded.		
INPUT	XL3	ARCHIVE DATE - Date for use in encoding the name. XL3'YYDDDD' (BINARY).		
INPUT	XL2	ARCHIVE TIME - Time for use in encoding the name. XL2'HHMM' (BINARY UNSIGNED).		
OUTPUT	xL44	SAMS:Disk RACF ENCODED PROFILE NAME		
OUTPUT	HW	Specify Return Code	H'1'	User Exit Encoded the Encoded PROFILE NAME
			H'2'	Continue With SAMS:Disk Encode Of Profile Name
			H'3'	Error, Reject Encode Request

RACFSMEX — Process RACF Protected DSN with Sequential Migrate

The module name supplied for this exit will receive control from the SAMS:Disk sequential migrate function when it needs to determine if a RACF-protected data set may be processed. When the exit receives control, SAMS:Disk has determined how to process the data set.

The user exit may elect to override the decision or accept it.

Use sysparm RACFSMEX to specify the module name to be invoked for this exit.

Table 5-44. Parameters Passed to RACFSMEX

Type	Size	Description		
INPUT	XL256	DATA SET IDENTIFIER - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	HW	COUNT OF VOLUMES DSN RESIDES ON		
INPUT	nCL6	LIST OF VOLUMES DSN RESIDES ON		
IN/OUT	HW	Decide To Process Data Set	H'1'	Process Racf-protected Data Set
			H'2'	Reject Data Set From Processing

RACHKAEX — RACF RACHECK Processing Post-RACHECK Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for RACHECK processing. It receives control after the RACHECK macro has been issued.

The exit may inspect the return code from the RACHECK macro and/or change it.

Use sysparm RACHKAEX to specify the module name to be invoked for this exit.

Table 5-45. Parameters Passed to RACHKAEX

Type	Size	Description		
INPUT	CL44	Data Set Name Checked		
INPUT	HW	Volume Count - A count of the volumes on which the data set resides. This count indicates how many volumes are contained in the data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which The Data Set Resides		
INPUT	CL1	Requested Access Level	'U'	Update access requested
			'R'	Read access requested
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
IN/OUT	HW	The return code from the RACHECK macro is placed in this area. If desired it may be changed to one of the following valid values:		
		H'0'	OK to Process Data Set	
		H'8'	User is not Authorized	
		H'4'	Resource Not Defined	
		H'16'	Critical Error	

RACHKPEX — RACF RACHECK Processing Pre-RACHECK Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for RACHECK processing. It receives control before the RACHECK macro is issued.

The exit may change RACHECK parameters, or cause the RACHECK to be bypassed and return its own return code.

Use sysparm RACHKPEX to specify the module name to be invoked for this exit.

Table 5-46. Parameters Passed to RACHKPEX

Type	Size	Description		
INPUT	CL44	Data Set Name To Be Checked		
INPUT	HW	Volume Count - A count of the number of volumes on which the data set resides. This count indicates how many volumes are contained in the data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which The Data Set Resides		
INPUT	CL1	Requested Access Level	'U'	Update access requested
			'R'	Read access requested
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
OUTPUT	HW	If the exit desires to perform the RACF check and to have SAMS:Disk bypass the RACHECK, place a return code in this parameter. To cause SAMS:Disk to continue processing, leave this parameter with the value as it was on entry.		
		H'0'	OK TO PROCESS DATA SET	
		H'8'	User is not Authorized	
		H'4'	Resource Not Defined	
		H'16'	Critical Error	

RADEFAEX — RACF RACDEF Processing Post-RACDEF Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for RACDEF processing. It receives control after the RACDEF macro has been issued.

The exit may inspect the parameters passed and returned from RACDEF processing.

Use sysparm RADEFAEX to specify the module name to be invoked for this exit.

Table 5-47. Parameters Passed to RADEFAEX

Type	Size	Description		
INPUT	CL44	Old Data Set Name - The data set used as the model to generate the RACF profile for the NEWNAME data set name.		
INPUT	HW	Old Volume Count - A count of volumes on which the data set resides. This indicates how many volumes are contained in data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which Old Data Set Resides		
INPUT	CL44	New Data Set Name - This is the data set that received the RACF profile.		
INPUT	HW	New Volume Count - A count of volumes on which the data set resides. This indicates how many volumes are contained in data for parameter 6.		
INPUT	nCL6	List Of Volumes On Which New Data Set Resides		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
OUTPUT	HW	The return code from the RACDEF macro is placed here. It may be changed by the exit. Use one of the following valid values:		
		H'0'	RACDEF Processed OK	
		H'8'	RACF Installation Exit Failed	
		H'4'	Resource Previously Defined	
		H'16'	Old Data Set not Defined	
		H'20'	Critical Error	

RADEFPEX — RACF RACDEF Processing Pre-RACDEF Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for RACDEF processing. It receives control before the RACDEF macro is issued.

Note: If you use the Fujitsu RACF name conversion exit to change nodes within a data set name, you must also use this exit to restore the name to its original form and return it as the first parameter defined below. The exit may change RACDEF parameters, or cause the RACDEF to be bypassed and return its own return code.

Use sysparm RADEFPEX to specify the module name to be invoked for this exit.

Table 5-48. Parameters Passed to RADEFPEX

Type	Size	Description		
IN/OUT	CL44	Old Data Set Name - This is the data set to be used as the model to generate the RACF profile for the NEWNAME data set name.		
INPUT	HW	Old Volume Count - A count of volumes on which the data set resides. This indicates how many volumes are contained in data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which Old Data Set Resides		
INPUT	CL44	New Data Set Name - This is the data set to be given a new RACF profile		
INPUT	HW	New Volume Count - A count of volumes on which the data set resides. This indicates how many volumes are contained in data for parameter 6.		
INPUT	nCL6	List Of Volumes On Which New Data Set Resides		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
OUTPUT	HW	If the exit desires to perform the RACF RACDEF and have SAMS:Disk bypass the RACDEF, place a return code in this parameter. To cause SAMS:Disk to continue processing, leave this parameter with the value as it was on entry.		
		H'0'	RACDEF Processed OK	
		H'8'	RACF Installation Exit Failed	
		H'4'	Resource Previously Defined	
		H'16'	Old Data Set not Defined	
		H'20'	Critical Error	

RADELAEX — RACF RACDEF (Delete) Processing Post-Processing

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for deleting RACF profiles. It receives control after the RACDEF macro has been issued.

The exit may inspect parameters passed to RACDEF and change the return code from the RACDEF macro.

Use sysparm RADELAEX to specify the module name to be invoked for this exit.

Table 5-49. Parameters Passed to RADELAEX

Type	Size	Description		
INPUT	CL44	Profile Name Deleted		
INPUT	HW	Volume Count - A count of volumes on which the data set resides. This count indicates how many volumes are contained in the data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which Data Set Resides		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
IN/OUT	HW	The RETURN CODE from RACDEF is placed in this parameter. It may be changed to any of the following values if desired.		
		H'0'	Profile Deleted	
		H'4'	Resource not Defined	
		H'8'	Failed by RACF Exit	
		H'16'	Critical Error	

RADELPEX — RACF RACDEF (Delete) Processing Pre-RACHECK Exit

The module name supplied for this exit will receive control from the SAMS:Disk module responsible for deleting RACF profiles. It receives control before the RACDEF macro is issued.

The exit may change RACDEF parameters, or cause the RACDEF to be bypassed and return its own return code.

Use sysparm RADELPEX to specify the module name to be invoked for this exit.

Table 5-50. Parameters Passed to RADELPEX

Type	Size	Description		
INPUT	CL44	Profile Name To Be Deleted		
INPUT	HW	Volume Count - A count of the number of volumes on which the data set resides. This count indicates how many volumes are contained in the data for parameter 3.		
INPUT	nCL6	List Of Volumes On Which Data Set Resides		
INPUT	CL1	Data Set Type	'V'	Request for VSAM cluster
			'N'	Request for non-VSAM data set
OUTPUT	HW	If the exit desires to perform the RACDEF and to have SAMS:Disk bypass the RACDEF, place a return code in this parameter. To cause SAMS:Disk to continue processing, leave this parameter with the value as it was on entry.		
		H'0'	Profile Deleted	
		H'4'	Resource not Defined	
		H'8'	Failed by RACF Exit	
		H'16'	Critical Error	

RECPRIEX — Screen Recover Requests

The module name supplied for this exit will receive control from the RECOVER proc before selecting data sets to recover. Different selection criteria can be applied to the RECOVER function based on the fields in the DSNINDEX record. Use sysparm RECPRIEX to specify the module name to be invoked for this exit.

Table 5-51. Parameters Passed to RECPRIEX

Type	Size	Description		
INPUT	XL256	DSNINDEX record for the data set to be recovered.		
OUTPUT	HW	Specify return code	H'0'	Continue Processing Data Set Through Normal Selection Criteria (command parms)
			H'1'	Select this Data Set and Bypass the Normal Selection Criteria (command parms)
			H'2'	Do not Select this Data Set, Bypass Normal Selection Criteria (command parms)

RESAFTEX — Post Restore Exit

The module name supplied for this exit will receive control from the RESTORE, DEFERRED RESTORE and RECOVER after restoring the data set or cluster. Use sysparm RESAFTEX to specify the module name to be invoked for this exit.

Table 5-52. Parameters Passed to RESAFTEX

Type	Size	Description				
INPUT	XL256	DSNINDEX record for the data set to be restored				
INPUT	XL140	Format-1 DSCB for restored data set				
INPUT	HW	Return code of restore	H'1'	Data Set Restored Okay		
			H'2'	Data Set Restored Okay, But Disposition Failed		
			H'4'	Data Set Restore Failed		
INPUT	XL4	Disposition indicators	CL1	Scratch	U	no scratch
					S	scratch
			CL1	Catalog	C	catalog
					N	no catalog
					R	recatalog
			CL1	Create Date	N	no reset of create date
					C	reset of create date
			XL1	Erase	x'00'	no erase
					x'01'	erase
INPUT	HW	Return code of disposition processing				
INPUT	CL8	Function issuing restore				
INPUT	HW	Count of volumes supplied in VOLUME parameter				
INPUT	nCL6	Volume(s) supplied in VOLUME parameter or blank				
INPUT	CL1	Auto-restore indicator	N	not an auto-restore		
			Y	is an auto-restore		
INPUT	CL1	TSS task indicator	N	not a TSS task		
			Y	is a TSS task		
INPUT	CL8	Jobname				

RESPRIEX — Screen Restore Requests

The module name supplied for this exit will receive control from RESTORE, DEFERRED RESTORE and RECOVER prior to restoring the data set or cluster. The user may request that SAMS:Disk reject the restore request or continue processing.

Use sysparm RESPRIEX to specify the module name to be invoked for this exit.

Table 5-53. Parameters Passed to RESPRIEX

Type	Size	Description				
INPUT	XL256	DSNINDEX record for the data set to be restored				
OUTPUT	CL50	Message Field - Place message here when return code is set to 2 to reject request. The message will be printed with SAMS:Disk output.				
OUTPUT	HW	Specify return code	H'1'	Continue Normal Processing		
			H'2'	Terminate Processing for Restore		
INPUT	CL8	Function issuing restore				
IN/OUT	CL44	New name for data set				
IN/OUT	HW	Count of volumes supplied in VOLUME parameter				
IN/OUT	nCL6	Volume(s) supplied in VOLUME parameter or blank. Current SAMS:Disk processing limits the volume list to one volume. Do not enter more than one.				
IN/OUT	CL8	Pool name specified on command or blank				
IN/OUT	XL4	Disposition indicators	CL1	Scratch	U	no scratch
					S	scratch
			CL1	Catalog	C	catalog
					N	no catalog
					R	recatalog
			CL1	Create date	N	no reset of create date
					C	reset of create date
			XL1	Erase	x'00'	no erase
					x'01'	erase
INPUT	CL1	Auto-restore indicator	N	not an auto-restore		
			Y	is an auto-restore		
INPUT	CL1	TSS task indicator	N	not a TSS task		
			Y	is a TSS task		
INPUT	CL8	Jobname				
INPUT	CL8	Password supplied in PASSWORD parameter				

RETAFTEX — Retention Control Data Set Screening (Post-validate)

The module name supplied for this exit will receive control from the SAMS:Disk retention control function after it has retrieved the format-1 DSCB and qualified it for processing. The disposition of the data set is shown in field (PARM 3).

The user may request that SAMS:Disk reject the data set for processing or continue on to process the data set as normal.

Use sysparm RETAFTEX to specify the module name to be invoked for this exit.

Table 5-54. Parameters Passed to RETAFTEX

Type	Size	Description
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.
IN/OUT	XI14	The DISPOSITION field to be used to process the data set. Portions of this field have been taken from the SAMS:Disk RETVOLSPEC table.
		1 — 4 TASC FIELD from RETVOLSPEC table entry
		1 Retention selection criteria that caused the data set to be selected for processing. Modification of this field will not affect any SAMS:Disk processing or output contents.
		2 — 4 The ASC values from the RETVOLSPEC table entry causing the data set to be processed. These values may be modified with valid ASC values to change the way in which SAMS:Disk will process the data set.
		5 RESERVED
		6 — 8 XL3'yyddd' Current Date (BINARY)
		9 — 10 RESERVED
		11 — 14 Archive Reason Codes - The four characters of reason codes that indicate on the RETAIN report why the data set was selected for processing. This field may be modified to reflect different values on the RETAIN report.

RETGDGEX — Return GDG Limit to Implicit Archival for a DSN

This module is given control after implicit archival has failed to obtain a generation limit from the GDGDASDG table for a GDG data set.

The user may supply the GDG limit for the data set.

Use sysparm RETGDGEX to specify the module name to be invoked for this exit.

Table 5-55. Parameters Passed to RETGDGEX

Type	Size	Description		
INPUT	CL44	DATA SET NAME		
OUTPUT	HW	GDG LIMIT COUNT to use for the data set		
OUTPUT	HW	Specify Return Code	H'1'	Generation Limit Returned
			H'2'	Use Default Generation Limit

RETPRIEX — Retention Control DSN Screening (Pre-validation)

The module name supplied for this exit will receive control from the SAMS:Disk retention control function after it has retrieved the format-1 DSCB from disk and prior to qualifying it for processing. The user may request that SAMS:Disk reject the data set for processing, continue on to qualify the data set for processing, and/or change the disposition action SAMS:Disk is to use for the data set.

Use sysparm RETPRIEX to specify the module name to be invoked for this exit.

Table 5-56. Parameters Passed to RETPRIEX

Type	Size	Description		
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.		
OUTPUT	XI14	DISPOSITION field. If the RETURN CODE is set to 3, positions 2 through 4 must contain valid ASC values to be used in place of the values normally taken from the RETVOLSPEC table.		
		1 — 4	TASC FIELD (like RETVOLSPEC table)	
		1	T value of TASC field. This value is set to N upon entry to the exit. Modification of this field will not affect any SAMS:Disk processing or output contents.	
		2 — 4	This field is initialized to C'NNN' upon entry to the exit. If the exit routine sets the value of parameter 4 to H'3', this field must be filled with valid ASC values to determine how to process the data set.	
		5	RESERVED	
		6 — 8	XL3'yyddd' Current Date (BINARY)	
		9 — 10	RESERVED (DO NOT CHANGE)	
		11 — 14	ARCHIVE REASON CODES - The 4 characters of reason codes that indicate on the RETAIN report why the data set was selected for processing. This field may be modified to reflect different values on the RETAIN report.	
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Data Set
			H'3'	Initiate Disposition Per Disposition Field

RLSEAFEX — Idle Space Release Screening Exit

The module name supplied for this exit is given control after a data set has been selected for processing by idle space release.

The user may request that SAMS:Disk either continue normal processing or bypass the data set. When VSAM data sets are being processed, the entire cluster will be bypassed if this exit determines that either the data or index component was bypassed.

Use sysparm RLSEAFEX to specify the module name to be invoked for this exit; the EXIT= parameter on the command will override this default value.

Table 5-57. Parameters Passed to RLSEAFEX

Type	Size	Description		
INPUT	CL256	Data Set Identifier - Format-1 DSCB and appendage for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL256	Volume Identifier - Format-4 DSCB and appendage for data set to be processed. Turn to page 573 for a description of this area. The volume serial is in positions 1 to 6.		
INPUT	XL1	Simulation Flag	X'01'	simulate mode
			X'00'	live mode
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Bypass Processing for Data Set
INPUT	CL44	Vsam Cluster Name - For non-VSAM data sets, this field is filled with blanks. For VSAM data sets, it contains the cluster name for the component passed in the format-1 DSCB and the format-4 DSCB.		

RPTDSBEX — Process Data Set Billing Report Records

The module name supplied for this exit will receive control from the SAMS:Disk billing EXTEND processing. It is passed the report record (see the mapping for report record format RPTNDSNDSB on page [479](#) in this manual).

When this exit is specified, the standard SAMS:Disk report record is not written.

Use sysparm RPTDSBEX to specify the module name to be invoked for this exit.

Table 5-58. Parameters Passed to RPTDSBEX

Type	Size	Description
INPUT	H'1'	OPTION - always a halfword of 1.
INPUT	CL6	VOLUME
INPUT	CL44	DATA SET NAME
INPUT	XL159	REPORT RECORD - See RPTNDSNDSB on page 479 for more information.

RPTDSNEX — Screen Data Sets to be Reported On

The module name supplied for this exit will receive control from the SAMS:Disk reports function prior to generating report records for any data set.

The user may request that SAMS:Disk reject the data set for processing, or continue on to report on the data set.

Use sysparm RPTDSNEX to specify the module name to be invoked for this exit.

Table 5-59. Parameters Passed to RPTDSNEX

Type	Size	Description		
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing for Data Set

RPTDSUEX — Data Set Utilization Report Record Screening

The module name supplied for this exit will receive control from the Data Set Utilization report generator. It is passed each record generated for the report.

The user may request that SAMS:Disk reject the record for printing or continue processing as normal.

Use sysparm RPTDSUEX to specify the module name to be invoked for this exit.

Table 5-60. Parameters Passed to RPTDSUEX

Type	Size	Description		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.		
INPUT	XL256	Data Set Identifier - Format-1 DSCB and appendage for data set to be processed. Turn to page 573 for a description of this area.		
IN/OUT	XL	DSU Report Record. Turn to page 487 for a complete description of this record.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Do not Print Line for this Record

RPTMVDEX — Multiple Volume Dictionary Record Screening

The module name supplied for this exit will receive control from the MVD report generator. It is passed each record generated for the report.

The user may request that SAMS:Disk reject the record for printing or continue processing as normal.

Use sysparm RPTMVDEX to specify the module name to be invoked for this exit.

Table 5-61. Parameters Passed to RPTMVDEX

Type	Size	Description		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 to 6.		
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
IN/OUT	XL	MVD Report Record. Turn to page 465 for a complete description of this record.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Do Not Print Line for this Record

SCRTCHEX — Screen SAMS:Disk Scratch Data Set Request

The module name supplied for this exit will receive control from any SAMS:Disk function requesting to scratch a user's data set. It is invoked before scratch processing has started.

The user may request that SAMS:Disk reject the request to scratch the data set, or continue to process it.

Use sysparm SCRTCHEX to specify the module name to be invoked for this exit.

Table 5-62. Parameters Passed to SCRTCHEX

Type	Size	Description		
INPUT	CL44	Data Set Name - Name of data set to be scratched.		
INPUT	HW	Volume Count - Count of volumes on which data set to be scratched resides.		
INPUT	nCL6	Volume List - A list of 6-character volume serials that contain the data set to be scratched. Parameter 2 contains a count of the number of volumes in this list.		
OUTPUT	CL100	Message - If desired, the exit routine may place a message here to be printed. The message will be printed if the return code parameter contains a 2 and the message area is non-blank.		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Reject Request to Scratch Data Set

SECURAEX — Security Validation Exit, Post-SAMS:Disk Decision

The module name supplied for this exit will receive control whenever SAMS:Disk security routines are called to validate access to a protected resource. This is the post-processing exit. It will receive control after SAMS:Disk has invoked the user security system or SAMS:Disk security module and has made a decision as to whether the data set can be processed by SAMS:Disk.

The exit routine has a chance to examine the decision made by SAMS:Disk and, optionally, override the decision. Note that if SAMS:Disk has rejected the data set for processing and messages are to be written to the user, the messages have already been written at this point. To intercept security before messages have been issued, see the SECURPEX (pre-processing) exit.

The parameters to this exit are exactly the same as to the pre-processing exit (SECURPEX) except that the return code field has been set to reflect the decision SAMS:Disk has made regarding processing of the data set. See the documentation

for SECURPEX for exit module parameter descriptions. The user exit may change the return code to any valid value before returning control to SAMS:Disk.

Use sysparm SECURAEX to specify the module name to be invoked for this exit.

SECURPEX — Security Validation Exit, Pre-SAMS:Disk Decision

The module name supplied for this exit will receive control whenever SAMS:Disk security routines are called to validate access to a protected resource. This is the pre-processing exit. It will receive control before SAMS:Disk invokes the user security system or SAMS:Disk security module to make a decision as to whether the data set can be processed by SAMS:Disk.

The exit routine has a chance to make the decision itself by setting the return code to one of the valid values. If the exit decides to leave the decision to SAMS:Disk, it can return without changing the return code.

Note: When security checking is being made for a data set that exists in the SAMS:Disk archives, the volume serial passed will be “*ARCH*”. This is not a real DASD volume. It has special meaning to SAMS:Disk.

Use sysparm SECURPEX to specify the module name to be invoked for this exit.

Table 5-63. Parameters Passed to SECURPEX

Type	Size	Description
INPUT	HW	Resource Type - The first parameter to this exit indicates the resource type for which security is being checked. It is in a halfword binary count format. The following list indicates the resource types protected and the value corresponding to that type:
		H'1' Data Set Name
		H'2' reserved
		H'3' Dasd Volume Serial
		H'4' Vsam Catalog Name
		H'5' SAMS:Disk Command
		H'6' Vsam Component (cluster) Name
		H'7' SAMS:Disk Subfile Of The Files Data Set
		H'8' Member Of SAMS:Disk Parmlib Data Set

Type	Size	Description
Parameters 2 through 4 passed to the exit are always the same regardless of the resource type for which they are being called. They are:		
INPUT	CL8	Access Request - This field contains an character identifier of the type of access being requested for the resource. The numbers following indicate what RESOURCES are passed the ACCESS TYPE. The following is a list of valid ACCESS TYPES:
		CL8'INPUT ' 1,3,4,6
		CL8'OUTPUT ' 1,4,5,6,7,8
		CL8'ALLOCATE' 1,6
		CL8'SCRATCH ' 1,6
		CL8'CATALOG ' 1,6
		CL8'RENAMEF ' 1,6
		CL8'RENAMET ' 1,6
		CL8'IN/OUT ' 1,6
INPUT	CL1	Message Indicator - This 1-character field indicates if SAMS:Disk has been requested to issue a message if the request to process the resource is rejected. Value Y indicates a message is to be issued. Value N indicates no messages are to be issued.
OUTPUT	HW	Return Code - This field indicates whether the request to process is to be accepted or rejected. The value is in halfword binary count format. It is initialized to zero. If not modified, SAMS:Disk will continue security checking for the resource and make the decision to accept or reject it. If the user exit routine makes the decision, this field is set to reflect it. The valid values for this field are:
		H'0' Let SAMS:Disk make the decision (SECURPEX ONLY)
		H'1' Let user process resource
		H'8' Reject request to process

Additional parameters for SECURPEX are Resource Type = 1 (Data Set Name) parameters below:

Table 5-64. Parameters 5 - 8 (DSNAME) for SECURPEX

Type	Size	Description
INPUT	XL256	Data Set Identifier - Format-1 DSCB and SAMS:Disk appendage for the data set to be processed. Turn to page 573 for a description of this area. Note: Reading the directory of a PDS requires security clearance that has not been performed yet. Therefore the PDS directory fields in the appendage area have not been filled at the time of this exit.
INPUT	HW	Volume Count - The number of volumes on which this data set resides.
INPUT	nCL6	List Of Volumes - A list of volumes on which the data set resides. The number of entries in this list depends on the count from parameter 6.
INPUT	CL 1	Password Indicator - A 1-character flag whose value indicates if password-protected data sets are to be processed. A Y indicates the user has requested to process password-protected data sets. An N indicates the user has not specifically requested for process password-protected data sets. If the PASSWORD parameter is available on the SAMS:Disk command, this indicates if it has been supplied or not.

Additional parameters for SECURPEX are Resource Type = 3 (DASD Volume Serial) described below:

Table 5-65. Resource Type = 3 (VOLSER) for SECURPEX

Type	Size	Description
INPUT	CL6	Volume Serial Name - The 1- to 6-character volume serial that the user is requesting to process.

Additional parameters for SECURPEX are Resource Type = 4 (VSAM Catalog Name) described below:

Table 5-66. Resource Type = 4 (VSAM Catalog Name)

Type	Size	Description
INPUT	CL44	Vsam Catalog Name - The 1- to 44-character data set name of the VSAM catalog.
INPUT	CL8	Catalog Password - If the user has supplied the PASSWORD for the catalog in the SAMS:Disk command, it will be present here.
INPUT	CL6	Catalog Volser - The volume serial name on which the VSAM catalog resides.

Additional parameters for SECURPEX are Resource Type = 5 (SAMS:Disk Command) described below:

Table 5-67. Resource Type = 5 (Commands) for SECURPEX

Type	Size	Description
INPUT	CL12	SAMS:Disk COMMAND - The 1- to 12-character name of the SAMS:Disk command the user is attempting to use. This is the full command name regardless of how it was entered.

Additional parameters for SECURPEX are Resource Type = 6 (VSAM Component) described below:

Table 5-68. Resource Type = 6 (VSAM Component)

Type	Size	Description
INPUT	CL44	VSAM Component Name - The 1- to 44-character data set name of the VSAM component to be accessed.
INPUT	CL8	VSAM Component Type - This field indicates the type of component for which the request applies. The valid values are:
INPUT	CL8	VSAM Password - The VSAM password specified for the component name.
INPUT	XI2	VSAM Security Flags - This is the VSAM security flags from the VSAM CATALOG record for the component. See the Fujitsu CATALOG MANAGEMENT PLM for a complete description of these flags.
INPUT	HW	Volume Count - A count of 1. This value never changes for the exit.
INPUT	HW	Catalog Volume Serial - The DASD volume serial that contains the VSAM catalog that describes this component.

Additional parameters for SECURPEX are Resource Type = 7 (SAMS:Disk Subfile) described below:

Table 5-69. Resource Type = 7 (Subfile) for SECURPEX

Type	Size	Description
INPUT	CL8	Subfile Name - The 8-character name of the subfile in the SAMS:Disk files data set for which access is being requested.
INPUT	CL8	Member Name - The 8-character name of the member of the parmlib data set for which access is being requested.

SPACECEX — Space Conversion Exit (Non-vsam)

The module name supplied for this exit will receive control from any SAMS:Disk function requesting allocation of DASD space for a non-VSAM data set.

The user may inspect or modify the number of tracks that SAMS:Disk has chosen to allocate for the data set.

Use sysparm SPACECEX to specify the module name to be invoked for this exit.

Table 5-70. Parameters Passed to SPACECEX

Type	Size	Description
IN/OUT	HW	New tracks to allocate for the data set
INPUT	CL44	Data set name being allocated
INPUT	CL6	Volume on which allocation will occur
INPUT	FW	Original Tracks for the data set on its original device type
INPUT	XL2	Original Source Device Type as expressed in the UCBTYPE field
INPUT	XL2	Target Device Type as expressed in the UCBTYPE field
INPUT	FW	Tracks Per Cylinder on SOURCE device type
INPUT	FW	Tracks Per Cylinder on TARGET device type
INPUT	FW	Original Blocksize
INPUT	FW	New Blocksize
INPUT	XL2	DSORG as defined in the VTOC entry
INPUT	XL1	RECFM as defined in the VTOC entry
INPUT	HW	KEY LENGTH for a key-sequenced data set
INPUT	XL1	SCALO field as defined in the VTOC entry
INPUT	XL1	OPTCD field as defined in the VTOC entry

SPFHDX — PFD Report Heading Exit

The module name supplied for this exit will receive control after SAMS:Disk has formatted the top two heading lines for an PFD report, but before it has printed them.

The user may change either or both of the report headings to suit installation standards. If the headings are not modified, SAMS:Disk will print the headings it has built.

Use sysparm SPFHDX to specify the module name to be invoked for this exit.

Table 5-71. Parameters Passed to SPFHDX

Type	Size	Description
INPUT	CL60	Report title user specified (do NOT modify)
IN/OUT	CL132	Top heading of PFD report
IN/OUT	CL132	Second heading of PFD report

SPFORDEX — PFD Online Reporting Report Definition Authorization

A specific exit can be invoked to send SAMS:Disk messages through the SAMS subsystem. The module name supplied for this exit will receive control if a user has selected a report definition option in PFD online reporting.

The exit routine is passed the TSS user ID of the user requesting the option and passes back a return code to indicate if the user is authorized or not.

Use sysparm SPFORDEX to specify the module name to be invoked for this exit.

Table 5-72. Parameters Passed to SPFORDEX

Type	Size	Description		
INPUT	CL8	TSS user ID		
OUT	H	Return code	H'1'	user is authorized
			other	user is not authorized

SPFORSEX — PFD Online Reporting Authorization

The module name supplied for this exit will receive control if a user has chosen to execute an online report.

The exit routine is passed the TSS user ID of the user requesting the report and passes back a return code to indicate if the user is authorized or not.

Use sysparm SPFORSEX to specify the module name to be invoked for this exit.

Table 5-73. Parameters Passed to SPFORSEX

Type	Size	Description		
INPUT	CL8	TSS user ID		
OUT	H	Return code	H'1'	user is authorized
			other	user is not authorized

SYSOUTEX — SYSOUT Exit

The module name supplied for this exit will receive control for every line of printed output being directed to CMDPRINT, MSGPRINT or SYSPRINT.

The user may instruct SAMS:Disk to either print or bypass the line of output. Suppression of heading lines may yield unpredictable results, however, in that line count and paging are affected. You also must not attempt to use any of the standard SAMS:Disk techniques to issue an additional message from your exit. This recursiveness will be detected, and the message will be written to the operator instead of its designated location. You may of course redirect the line of output to any new target, such as a disk data set or a special operator console.

Unlike other SAMS:Disk exits, the module name to be invoked for this exit must be superzapped into the SAMS:Disk output module as follows:

```
NAME ADSUT190 SYSOUTEX
VER 0000 40404040,40404040
REP 0000 xxxxxxxx,xxxxxxx
```

where xxxxxxxx,xxxxxxx is the EBCDIC representation of the module name. If the module name is less than 8 characters, left justify and pad with blanks.

Warning: You must reapply this zap each time you install a new release of SAMS:Disk if that new release will overlay module ADSUT190.

Table 5-74. Parameters Passed to SYSOUTEX

Type	Size	Description		
INPUT	CL8	DDNAME (CMDPRINT, MSGPRINT or SYSPRINT)		
INPUT	CLxx	Print Line		
INPUT	CL1	Carriage Control for the print line:	C' '	single space
			C'0'	double space
			C'-'	triple space
			C'1'	new page, first line
INPUT	HW	LENGTH of the PRINT LINE		
INPUT	XL2	TYPE of PRINT LINE:	X'0000'	regular data line for printing
			C'H'—C'H9'1'	a heading line
			C'F1'—C'F9'	a footing line
OUTPUT	HW	Specify The Return Code	H'1'	continue (print the line)
			H'4'	skip the print line

TAOPPOEX — Archive Tape Processing Exit (Post-Open)

The module name supplied for this exit will receive control from SAMS:Disk functions that process archive tape data. It is invoked immediately after an archive tape volume has been opened.

The user exit allows the user to inspect and/or change the system control blocks related to the open.

Use sysparm TAOPPOEX to specify the module name to be invoked for this exit.

Table 5-75. Parameters Passed to TAOPPOEX

Type	Size	Description		
INPUT	CL8	SAMS:Disk Function - Indicates the SAMS:Disk function invoking the exit (that is, archive, restore, merge)		
INPUT	CL6	Type Of Open	“input”	indicating read
			“output”	indicating write
INPUT	CL1	Type Of Tape	M	indicates primary archive tape being processed
			C	indicates copy archive tape being processed
INPUT	XL	DCB - Data Control Block used for the open		
INPUT	XL	JFCB - Job File Control Block used for the open		

TAOPPREX — Archive Tape Processing Exit (Pre-Open)

The module name supplied for this exit will receive control from SAMS:Disk functions that process archive tape data. It is invoked prior to opening an archive tape volume.

The user exit allows the user to inspect and/or change the system control blocks related to the open.

Use sysparm TAOPPREX to specify the module name to be invoked for this exit.

Table 5-76. Parameters Passed to TAOPPREX

Type	Size	Description		
INPUT	CL8	SAMS:Disk FUNCTION - Indicates the SAMS:Disk function invoking the exit (that is, archive, restore, merge)		
INPUT	CL6	Type Of Open	"input"	indicating read
			"output"	indicating write
INPUT	CL1	Type Of Tape	M	indicates primary archive tape being processed
			C	indicates copy archive tape being processed
INPUT	XL	DCB - Data Control Block to be used when opening the tape volume. It is set up to open when the exit is called		
INPUT	XL	JFCB - Job File Control Block to be used when opening the tape. It is also set up for the open		

TAPCTLEX — Tape Management System Control Interface Exit

This exit is identical to TMSCTLEX. The module name supplied for this exit enables users to run more than one tape management system during a conversion process. See user exit TMSCTLEX for a full description, and a complete list of input and output parameters available.

TAPEDLEX — Archive Volume Being Deleted from Archives

The module name supplied for this exit will receive control from any SAMS:Disk function that is deleting TAPE/CARTRIDGE archive volumes from the archive index.

The archive volume record passed to this exit is always sequence #1. Parm 2 indicates the total number of archvols records in the list and Parm 3 is the address of the list of arckeys for this archive data set. (For example, if a multivolume archive data set is being deleted, parm 2 will be GT H' 1' and parm 3 will have more than one arckey in the list.)

No return code is accepted by this exit. SAMS:Disk processing cannot be altered.

Use sysparm TAPEDLEX to specify the module name to be invoked for this exit.

This exit is bypassed in SIMULATE mode.

Table 5-77. Parameters Passed to TAPEDLEX

Type	Size	Description
INPUT	XL110	Archvols Record - This is the archive volume index record from the files data set. ARCKEY is in positions 1 - 6. The true volser is in position 12 - 17. Turn to page 525 for a complete description of this record.
INPUT	H	List Count - Number of archive keys in the list that follows (maximum of 35 for any given multivolume archive data set).
INPUT	AL4	List Address - The address of the list of archive keys for the specified archive data set. (The number of keys in the list is specified in parm 2).

TMSCTLEX — Tape Management System Control Interface Exit

The module name supplied for this exit receives control from any SAMS:Disk function that is updating the status of an existing archive tape volume. This exit may then make the appropriate updates in whatever tape management data base is used to control the tapes. When SAMS:Disk changes the status of a tape volume, only the files data set is updated. This exit must provide the interface to the tape management system.

Note: User Exit TAPCTLEX enables support for sites that use more than one tape management system.

There are two options that correspond to the two reasons for an update:

1. SAMS:Disk no longer needs the tape. The tape can be returned to scratch status within the tape management system data base. This option is invoked when SAMS:Disk is going to expire or delete an archive volume.
2. The expiration date for a tape must be updated. This option is invoked when SAMS:Disk is going to alter the expiration date of an archive volume. When option 2 is invoked, a seventh parameter is passed. This parameter is the new expiration date that SAMS:Disk is going to reset for the archive volume. This parameter does not exist for option 1.

No messages are issued based on the return code passed back from this exit. The exit must provide these messages. The exit must also ensure the integrity of the tape management system data base and ensure that tapes are not prematurely expired.

Use sysparm TMSCTLEX to specify the module name to be invoked for this exit.

Table 5-78. Parameters Passed to TMSCTLEX

Type	Size	Description		
INPUT	H	OPTION:	1	Expire Tape
			2	Reset Expiration Date
INPUT	CL44	Tape DSN		
INPUT	H	Number of tape volumes Will always be 1		
INPUT	CL6	Tape volume		
OUTPUT	H	Return Code:	1	Update successful
			2 or more	update failed
OUTPUT	H	Reason Code:	Not used at this time. Can be set to any HW value.	
INPUT	PL4	New EXPDT - (option 2 only)		

Although you may choose to write this exit yourself, two exit programs are supplied as part of the SAMS:Disk system. This first program, "ADSTH014", is the interface to the CA1 External Data Manager (EDM) Interface which is supported in Release 4.8 or higher of CA1. The second program, "ADSTH025", is a direct interface for Release 4.7 and below of CA1.

Using program "ADSTH014" is the preferred method of interfacing with CA1. Instructions for implementing the External Data Manager Interface support can be found in the topic *"Method 1 — Controlling Tapes Via the EDM"* on page 63 of the *Installation Guide*.

Use of program "ADSTH025" is documented below.

The SAMS:Disk Direct Interface to CA1:

For expiration date control within a CA1 (release 4.7 and below) environment, we recommend that you install the direct interface that SAMS:Disk provides to CA1. Review the information presented below, as well as the topic *"Method 2 — Controlling Tapes by Expiration Date"* on page 65 of the *Installation Guide*, and then decide if the direct interface to CA1 is applicable and desirable in your installation. If it is not, you should change the expiration date being assigned via sysparm DYNEXPDT (turn to page 146 for a description of this sysparm) or your JCL to specify either a true expiration date or the value 99000, which can then be used to place the tapes under catalog control. Also see *"Assigning Tape Expiration Dates"* on page 42 for other applicable rules. The SAMS:Disk direct interface to CA1 uses a central module designed to handle the interchange and update of records in the TMC (Tape Management Catalog). All precautions are taken to ensure that updating is thorough and complete. However, if you use a CA1 release that is higher than 4.7 and specify sysparm DYNEXPDT to a value other than the default, you should be aware of the following limitations when using this interface:

1. Like CA1 Release 4.7 and below, this interface does not support expiration dates beyond the year 2000.
2. This interface does no special processing for expiration dates which fall within the ranges of cycle or frequency control.
3. This interface does not read or write exception or journal records.

When SAMS:Disk determines that an archive or backup tape can be returned to scratch status, it can inform CA1 in two different ways:

1. SAMS:Disk can retrieve the TMC record for that tape and set its expiration date to the current date. The tape is made a scratch when the CA1 scratch and clean jobs are run.
2. SAMS:Disk can take the same action as above, but also turn on the scratch bit indicator flag. Turning on this bit makes the tape an immediate scratch. This is particularly useful when also using the SAMS:Disk tape pool support.

Use of the direct interface to CA1 is not required. It simply provides the advantage of immediate updates to the CA1 tape management catalog when SAMS:Disk releases a tape volume from the archives or increases the expiration date of an archived data set. It has no effect during data set archival/backup processing since standard CA1 hooks in tape OPEN get the expiration date from system control blocks at that time.

Installing the SAMS:Disk Direct Interface to CA1:

1. Assemble module ADSTH012. The module is located in the installation library, and is written in Assembler Language. Use a standard assembly procedure, ensuring that the SAMS:Disk macro library is concatenated before all other fmaclibs in the SYSLIB statements. Also make sure that the CA1 macro library is in the SYSLIB statements but after the SAMS:Disk macro library.

If you have installed SAMS:Disk with SMP/E, this assembly should be done using SMP/E in the form of a usermod. Installation library member USERMODA can accomplish this for you. If you did not install SAMS:Disk with SMP/E, sample assemble/link JCL for assembling this module is located in installation library member INSTTMS.

We recommend using Assembler H to assemble SAMS:Disk modules. However, we will try to allow source members in the installation library to be assembled with Assembler F. If you use Assembler F, you must add

```
,AMODE=,SPLEVEL=
```

to the end of the PROC macro statement in your assemble source. If Assembler F cleanly assembles a SAMS:Disk module (condition code 0—no assembly errors) the module is correctly assembled. The problems with Assembler F are in the form of assembly errors (for example, “IF STATEMENT OUT OF ORDER. . .”) not experienced with Assembler H.

Note that the CA1 macros may change with each release of CA1. Therefore, you may need to assemble and link the SAMS:Disk module ADSTH012 when you install a new release of CA1.

Note: If a problem arises because some of the needed CA1 modules cannot be found, simply put a PRINT GEN statement in module ADSTH012 and reassemble it. The assembly listing with the expanded macros will identify the CA1 modules that must be found to complete a successful installation.

2. Verify that the module is in the SAMS:Disk load library and there are no unresolved references in the link-edit.

Activating the SAMS:Disk Direct Interface to CA1:

1. Make the CA1 load library available to SAMS:Disk. Either put the CA1 load modules in a LINKLIST library, or put the CA1 load library in the STEPLIB for all SAMS:Disk backup, archive, restore and IXMAINT tasks.
2. Review the following sysparms in the *Sysparms* section:

TPMGMTX — on page [192](#)
TMSCTLEX — on page [191](#)
UCC1SCRT — on page [194](#)
UCC1EXPR — on page [194](#)
UCC1BYPS — on page [193](#)
UCC1EXTN — on page [194](#)
3. Turn on the CA1 support by specifying sysparm TMSCTLEX with a value of ADSTH025.
4. Decide which of the two methods of scratching tapes SAMS:Disk is to use. SAMS:Disk defaults to just setting the expiration date to the current date. If you want SAMS:Disk to also set the scratch bit, causing CA1 to immediately treat it as a scratch tape, specify sysparm UCC1SCRT with a value of Y.

SAMS:Disk/CA1 Interfacing Considerations

We recommend that you run the DSNDELETE command of IXMAINT (turn to page [310](#) of the *User's Guide* for details) nightly before the CA1 scratch and clean functions are run. This helps to ensure that CA1 is always current with SAMS:Disk.

If a job abends while writing a tape and you are not using the EDM Interface as your SAMS:Disk interface, CA1 defaults to considering that output tape a scratch. To SAMS:Disk, however, the partial tape is a good tape and must be kept. You should take special precautions to prevent CA1 from marking SAMS:Disk output tapes as scratch tapes after an abend. This can be done manually, or you can make use of one of the exits within the CA1 system. (A SAMS:Disk user has supplied a sample for this exit in member SLI035 of the user mod library. You should consult your local CA1 support staff or Computer Associates directly, if necessary, if you have further questions regarding their exit.)

TPMGMTX — Tape Management Interface Exit

The module name supplied for this exit will receive control from any SAMS:Disk function that is updating the status of an existing archive volume. Typical types of updates are if SAMS:Disk is going to expire or delete archive volumes or if SAMS:Disk is going to update the expiration date of an archive volume.

There are two options possible for this exit. Option 1 is invoked when SAMS:Disk is going to expire or delete an archive volume. Option 2 is invoked when SAMS:Disk is going to alter the expiration date of an archive volume. When option 2 is invoked, a seventh parameter is given. This parameter is the new expiration date that SAMS:Disk is going to reset for the archive volume. This parameter does not exist for option 1.

If you have sysparm TMSCTLEX specified with “ADSTH014”, (that is you are using the External Data Manager facility of CA1 as your tape management interface), you should be aware that there is no facility available to change the expiration date within the CA1 database, therefore no processing will be done by SAMS:Disk. This user exit will still be called with this option, however, to support other tape management interfaces that may exist.

Use sysparm TPMGMTX to specify the module name to be invoked for this exit.

Table 5-79. Parameters Passed to TPMGMTX

Type	Size	Description		
INPUT	H	OPTION:	1	Expire Tape
			2	Reset Expiration Date
INPUT	CL44	Tape DSN		
INPUT	H	Number of tape volumes. Will always be 1.		
INPUT	CL6	Tape volume		
OUTPUT	H	Return Code	1	OK to update
			2 or more	stop update
OUTPUT	H	Reason Code	Not used at this time. Can be set to any HW value.	
IN/OUT	PL4	New EXPDT - (option 2 only).		

UNLOADEX — Process Unloaded Files Data Set Record

The module name supplied for this exit will receive control after a record has been selected for unload in the files unload process. The entire record is passed to the module and the user may modify any field desired. If the length of any record is altered, the user must ensure that the FILEDEFN table and files data set entries conform to the change. The exit may also be used to selectively eliminate records by setting a return code of 2.

Note: When record length is referenced, it means the file-defined record length plus decimal 12; for example, if a DSNINDEX record is passed and its file-defined length is 256, the length passed is 268.

Table 5-80. Parameters Passed to UNLOADEX

Type	Size	Description		
IN/OUT	CL8	Subfile Name		
IN/OUT	CL256	Files Record (maximum length of 256 bytes)		
IN/OUT	HW	Record Length (may be modified)		
OUTPUT	HW	Return Code	H'1'	Unload the record
			H'2'	Bypass the record

USERSENQ — ENQ/DEQ Exit

The module name supplied for this exit will receive control for each ENQ, DEQ, RESERVE or RELEASE to be issued. Its intended use is to allow the user to monitor, modify or replace the normal SAMS:Disk ENQ processing. It does not affect processing specified by the SAMS:Disk sysparms SPFENQQN, SPFENQTY or SPFENQWT. (To perform its intended task, the SAMS:Disk auto-restore function internally assigns and uses the module name “DMSNOENQ” for this user exit.)

A return code set by the exit module indicates what further action SAMS:Disk is to take.

Use sysparm USERSENQ to specify the module name to be invoked for this exit.

Table 5-81. Parameters Passed to USERSENQ

Type	Size	Description		
IN/OUT	CL2	Type Of Enq Request	CL2'E '	ENQ Exclusive
			CL2'S '	ENQ Shared
			CL2'C '	Change Shared to Exclusive
			CL2'D '	DEQ
			CL2'RE'	RESERVE and ENQ Exclusive
			CL2'RS'	RESERVE and ENQ Shared
			CL2'RC'	RESERVE and Change Shared to Exclusive
			CL2'RD'	RELEASE and DEQ
			CL2'TE'	Tape ENQ
			CL2'TD'	Tape DEQ
INPUT	CL44	DSNAME (the resource name)		
INPUT	CL6	VOLUME (for RESERVE requests)		
OUTPUT	H	Return Code	H'0'	Skip normal ENQ processing. DO NOT perform ENQ/DEQ processing.
			H'4'	Continue with normal SAMS:Disk ENQ/DEQ processing per the request type in parameter 1
			H'5'	Reject the ENQ/DEQ request

USERVENQ — VSAM DISP=SHR/OLD Exit

The module name supplied for this exit will receive control for each VSAM cluster to be allocated in a task. Its intended use is to allow the user to monitor or replace the DISP=SHR or DISP=OLD value.

A return code set by the exit module indicates what further action SAMS:Disk is to take.

Use sysparm USERVENQ to specify the module name to be invoked for this exit.

The USERVENQ user exit is not called for empty VSAM clusters. SAMS:Disk obtains all pertinent information from the catalog thereby removing the necessity of an enqueue.

Table 5-82. Parameters Passed to USERVENQ

Type	Size	Description		
IN/OUT	CL2	Type Of Enq Request	CL2'E '	ENQ Exclusive (DISP=OLD)
			CL2'S '	ENQ Shared (DISP=SHR)
INPUT	CL44	DSNAME (the resource name)		
INPUT	CL6	VOLUME		
OUTPUT	H	Return Code Value	H'0'	Continue with normal SAMS:Disk processing. Use the ENQ type returned in the first parameter.
			H'5'	Skip normal SAMS:Disk processing. Bypass processing of the cluster.

USRCODEX — User Code Substitution Exit

This module is invoked by DASD billing just prior to a table lookup to obtain a user code for a given data set name. The data set name is in a work area and may be modified by the exit module. If the data set name is modified, it will be used to search the USERCOTBL.

Use sysparm USRCODEX to specify the module name to be invoked for this exit.

Table 5-83. Parameters Passed to USRCODEX

Type	Size	Description		
INPUT	CL44	DSNAME For Data Set		
INPUT	CL12	Usercode Value		
OUTPUT	HW	Specify Return Code	H'1'	Obtain Code From Table and Continue
			H'2'	Use User Code Returned by Exit Module

VCCOPYEX — Move/Copy Screening

This module is invoked just prior to the copying a data set. The exit is the equivalent of the VCDMIGEX exit except it is invoked for the COPY command instead of the MIGRATE command. The exit may override the migration attempt.

Use sysparm VCCOPYEX to specify the module name to be invoked for this exit.

Table 5-84. Parameters Passed to VCCOPYEX

Type	Size	Description		
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 - 6.		
INPUT	XL16	Option Indicators - 16 one-character fields that indicate the options active at the time the exit is invoked. If the byte contains X'00', the option is not active. If it contains X'01', the option is active.	XL1	SIMULATE
			XL1	NOCATALOG
			XL1	NOSCRATCH
			XL1	FILL
			XL1	REDUCE
			XL1	PREALLOC
			XL1	CYLINDER
			XL1	PASSWORD
			XL1	ABSOLUTE
			XL1	TRACK
			XL1	CATALOGALL
			XL1	CONTIG
			XL1	CREATE
			XL1	reserved for future expansion
			XL1	reserved for future expansion
			XL1	reserved for future expansion
		The VCFLAGS macro is available in the SAMS:Disk macro library to map these flags		
OUTPUT	HW	Specify Return Code	H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Data Set

VCDMIGEX — Move/Copy Screening

This module is invoked just prior to the relocation of a data set. All selection screening by SAMS:Disk has been done and the exit is given a chance to override the migration.

Use sysparm VCDMIGEX to specify the module name to be invoked for this exit.

Table 5-85. Parameters Passed to VCDMIGEX

Type	Size	Description		
INPUT	XL256	Data Set Identifier - Format-1 DSCB, and appendage, for data set to be processed. Turn to page 573 for a description of this area.		
INPUT	XL140	Volume Identifier - Format-4 DSCB for data set. Volume serial is in positions 1 - 6.		
INPUT	XL16	Option Indicators - 16 one-character fields that indicate what options are active at the time the exit is invoked. If the byte contains X'00', the option is not active. If it contains X'01', the option is active.	XL1	SIMULATE
			XL1	NOCATALOG
			XL1	NOSCRATCH
			XL1	FILL
			XL1	REDUCE
			XL1	PREALLOC
			XL1	CYLINDER
			XL1	PASSWORD
			XL1	ABSOLUTE
			XL1	TRACK
			XL1	CATALOGALL
			XL1	CONTIG
			XL1	CREATE
			XL1	reserved for future expansion
			XL1	reserved for future expansion
			XL1	reserved for future expansion
OUTPUT	HW	Specify Return Code	The VCFLAGS macro is available in the SAMS:Disk macro library to map these flags	
			H'1'	Continue Normal Processing
			H'2'	Terminate Processing For Data Set

VSACESEX — Change VSAM Read/Write Processing Options

The module name supplied for this exit will receive control prior to every open of a VSAM data set. The purpose of the exit is to allow you to change the method SAMS:Disk will use to read VSAM clusters. It can also specify whether the format of the archived cluster should be in logical records or in control interval image copy format (ESDS clusters only). The values passed to the exit are those that SAMS:Disk has selected, based on sysparms in effect. If not modified, the values passed will be used when SAMS:Disk processes the cluster.

This exit would be used typically when your installation has either a systems or application program that formats VSAM entry-sequenced clusters (ESDS) in a non-standard format. These data sets must be archived in a control interval image copy format, so that they may be restored in the same manner. If you can recognize these data sets based upon data set naming conventions, this exit can be used to select image format for their archival, while all other data sets would use logical record format. The only advantage of using logical record format is that the control interval size of the data set can be changed when restoring the data set, whereas a control interval image copy must be restored to the same control interval size. If this flexibility is not required, you may specify sysparm VSARCFMT with a value of C to force all ESDS clusters to be archived in image copy format, and this exit would not be required.

Use sysparm VSACESEX to specify the module name to be invoked for this exit.

Table 5-86. Parameters Passed to VSACESEX

Type	Size	Description		
INPUT	CL44	Data Set Name		
INPUT	CL1	Cluster Type	'K'	Key-sequenced data set
			'E'	Entry-sequenced data set
			'R'	Relative record data set
IN/OUT	CL1	VSAM Read Access Type	'C'	Read the cluster with control interval processing
			'L'	Read the cluster with logical record processing
IN/OUT	CL1	Archive Format Of Data	'L'	Logical record format
			'C'	Control interval image copy format
Notes: 1. Format type C will be honored only for ESDS clusters.				
2. If you specify format type C but specify access type L (parameter 3), SAMS:Disk will change the access type to C.				
OUTPUT	HW	Result Code	0	Process the data set
			Non-zero	bypass the data set

VSALLOEX — Change VSAM Space Allocation Amounts

The module name supplied for this exit will receive control prior to all defines of VSAM components. The purpose of the exit is to allow the user to inspect the allocation amounts to be used in the DEFINE and change these values if desired. The exit not only receives the original allocation amounts used for the DEFINE, but also the new allocation amount that SAMS:Disk has calculated. It will be invoked for each component of the cluster.

The exit can either change the new allocation amount calculated by SAMS:Disk, or leave it alone, in which case that value will be used as the allocation amount. There is no RETURN CODE from this exit.

Note: This exit will not receive control if the allocation amounts for the cluster are specified on the RESTORE command.

Use sysparm VSALLOEX to specify the module name to be invoked for this exit.

Table 5-87. Parameters Passed to VSALLOEX

Type	Size	Description		
INPUT	XL7	ORIGINAL ALLOCATION AMOUNT		
		XL3	Primary Allocation Amount	
		XL3	Secondary Allocation Amount	
		X	Allocation Type	
		1	Allocation Is In Tracks	
		11	Allocation Is In Cylinders	
IN/OUT	XL7	New Allocation Amount (calculated By SAMS:Disk)		
		XL3	Primary Allocation Amount	
		XL3	Secondary Allocation Amount	
		X	Allocation Type	
		1	Allocation Is In Tracks	
		11	Allocation Is In Cylinders	
INPUT	F	Total Of All Allocated Extents In Original Data Set (total Is Always In Tracks)		
INPUT	F	Total Amount Of Used Space (total Number Of Tracks Actually Required By Original Data Set)		
INPUT	XL4	Original Device Type		
INPUT	XL4	New Device Type		
INPUT	F	High Used RBA Of Original Data Set		
INPUT	F	High Allocated RBA Of Original Data Set		
INPUT	F	Physical Block Size		
INPUT	F	Control Interval Size		
INPUT	CL1	Component Type	'D'	Data Component
			'I'	Index Component
INPUT	CL44	Data Set Name		
INPUT	H	Number Of Old Volumes		
INPUT	20CL6	List Of Old Volumes		
INPUT	H	Number Of New Volumes		
INPUT	20CL6	List Of New Volumes		

VSATTREX — VSAM Bufferspace and CI Size Exit

The module name supplied for this exit will receive control for each VSAM cluster to be defined in a task. Its intended use is to allow the user to monitor or replace the BUFFERSPACE, CISZ, and certain data set attributes prior to any DEFINE issued by SAMS:Disk.

Use sysparm VSATTREX to specify the module name to be invoked for this exit.

Table 5-88. Parameters Passed to VSATTREX

Type	Size	Description		
INPUT	CL44	Data Set Name		
INPUT	CL1	Component Type	CL1'D'	Data Component
			CL1'I'	Index Component
IN/OUT	F	Buffer Space Amount		
		Note: Value can be changed to specific amount on output, or can be set to zero to let AMS calculate new value.		
IN/OUT	F	Control Interval Size		
		Note: Value can be changed to specific amount on output, or can be set to zero to let AMS calculate new value.		
IN/OUT	XL2	Data Set Attribute Bytes		
		XL1	Data Set Attribute Byte (ATTR1)	
		1... ..	Use Speed Option When Loading Data Set	
		0... ..	Use Recover Option When Loading Data Set	
		.1... ..	Define The Data Set As Unique	
		.0... ..	Define The Data Set As Suballocated	
		..1... ..	Define The Cluster As Reusable	
		..0... ..	Define The Cluster As Not Reusable	
		...X XXXX	User Exit Should Not Modify These Flags	
		XL1	Data Set Sharing Attributes (ATTR2)	
		00... ..	Data Set Can Be Shared By Read Users Or By One Update/Output User	

Type	Size	Description
		01.. Data Set Can Be Shared By Read Users And By One Update/Output User
		10.. Data Set Can Be Fully Shared
		11.. Data Set Can Be Fully Shared With VSAM's Assistance
		Data Set Sharing Attributes Across Systems, As follows:
		..00 RESERVED
		..01 RESERVED
		..10 Data Set Can Be Fully Shared
		..11 Data Set Can Be Fully Shared With VSAM's Assistance
	 XXXX User Exit Should Not Modify These Fields

VSMAFTEX — Implicit VSAM Data Set Screening (Post-Validate)

The module name supplied for this exit will receive control from the implicit VSAM support after it has retrieved all information from catalog management regarding the cluster, and also after all screening tests are applied. If the VSMPRIEX user exit is also specified, this exit gets control after the VSMPRIEX exit has screened the request, regardless of the result returned by that exit. The result code passed as input represents the action that SAMS:Disk will take. This exit will also get control when processing requests that scan volumes using the SCAN command. As of Release 8.0, this is limited to only the PFD reporting function. When a data set name is supplied as a result of a SCAN REAL command, the catalog name passed as the fourth parameter will be nulls.

The user may request that SAMS:Disk reject or process the request, regardless of what value was found in the result code parameter. Use sysparm VSMAFTEX to specify the module name to be invoked for this exit.

Table 5-89. Parameters Passed to VSMAFTEX

Type	Size	Description		
IN/OUT	HW	Action Code. This parameter is both an input and an output parameter. The value on input informs the exit what action SAMS:Disk will take if unmodified by the exit. The value on output (if changed) instructs SAMS:Disk to take the new action.		
		H'0'	Data set will be processed	
		H'8'	Data set will be bypassed	
INPUT	CL8	Function Code	'REPORT'	report on the component
			'DELETE'	delete the component
			'BACKUP'	backup the component
			'ARCHIVE'	archive the component
INPUT	CL44	Data Set Name		
INPUT	CL6	Volser - primary volume of data component		
INPUT	CL44	Catalog Name - the catalog name in which the component is defined (if catalog scan)		
INPUT	CL1	DSNTYPE	'C'	Cluster component
			'I'	Index component
			'D'	Data component
			' '	Component type unknown
INPUT	CL6	Volser - volume serial number of this component		

VSMPRIEX — Implicit Data Set Screening (Pre-Validate)

The module name supplied for this exit will receive control from the implicit VSAM support after it has retrieved all information from catalog management regarding the cluster, but before any screening tests are applied. If the VSMAFTEX user exit is also specified, it will get control after the VSMPRIEX exit has screened the request, regardless of the result returned by this exit.

This exit will also get control when processing requests that scan volumes using the SCAN command. This is limited to only the PFD reporting function. When a data set name is supplied as a result of a SCAN REAL command, the catalog name passed as the fourth parameter will be nulls.

The user may request that SAMS:Disk continue with normal processing, reject the request, or unconditionally process the request.

Use sysparm VSMPRIEX to specify the module name to be invoked for this exit.

Table 5-90. Parameters Passed to VSMPRIEX

Type	Size	Description
OUTPUT	HW	Result Code. This parameter instructs SAMS:Disk as to what action to take. Its value on input is always 1, so if this module does not modify its value, normal processing will be carried out.
		H'1' Continue normal processing. This means that if any SELECT and/or EXCLUDE commands were specified, those tests will be applied after this exit returns control to SAMS:Disk.
		H'2' Terminate processing for the data set. Note that if the VSMAFTEX is also specified, that exit will still get control and can reverse the rejection request. Any SELECT and/or EXCLUDE commands are ignored.
		H'3' Unconditionally select the data set for processing. Note that if VSMAFTEX is also specified, that exit will still get control and can reverse the selection request. Any SELECT and/or EXCLUDE commands will be ignored.

Type	Size	Description		
INPUT	CL8	Function Code	'REPORT'	report on the component
			'DELETE'	delete the component
			'BACKUP'	backup the component
			'ARCHIVE'	archive the component
INPUT	CL44	Data Set Name		
INPUT	CL6	Volser - primary volume of data component		
INPUT	CL44	Catalog Name - the catalog name in which the component is defined (if catalog scan)		
INPUT	CL1	DSNTYPE	'C'	Cluster component
			'I'	Index component
			'D'	Data component
			' '	Component type unknown
INPUT	CL6	Volser - volume serial number of this components		

VSMSPWEX — VSAM Master Password Exit

The module name supplied for this exit will receive control after the SAMS:Disk master password is extracted. Its intended use is to allow the user to monitor or replace the SAMS:Disk master password with a user-supplied password.

Use sysparm VSMSPWEX to specify the module name to be invoked for this exit.

Table 5-91. Parameters Passed to VSMSPWEX

Type	Size	Description
INPUT	CL8	Function Being Executed
IN/OUT	CL8	SAMS:Disk Master Password

VSNEWNEX — VSAM New Name Exit

The module name supplied for this exit will receive control prior to every DEFINE of a VSAM component issued by SAMS:Disk. It is invoked whether or not a NEWNAME parameter was specified on the input command. Its intended use is to allow the user to monitor or replace the data set name for each component.

Use sysparm VSNEWNEX to specify the module name to be invoked for this exit.

Table 5-92. Parameters Passed to VSNEWNEX

Type	Size	Description		
INPUT	CL1	Component Type	C'C'	Cluster Component
			C'D'	Data Component
			C'I'	Index Component
INPUT	CL44	Base Component's Original Data Set Name		
INPUT	CL44	Base Component's New Name		
INPUT	CL44	Current Component's Data Set Name		
IN/OUT	CL44	New Data Set Name For Component		
IN/OUT	H	Return Code. Note that this is both an input and output parameter. If the value on input is greater than one, SAMS:Disk will let AMS generate a data set name for the component. If the value is less than or equal to one, the name passed as input will be used in the DEFINE. This value can be reset by the exit to either let AMS generate a name, or to indicate that the exit is supplying a name.		
		'1'	Use data set name supplied	
		'4'	Let AMS generate a component name	

XCPARCEX — XCOPY Archive Volume Selection Exit

This module is called during XCOPY processing to allow selection processing based on criteria other than that available through SELECT and XCOPY parameters.

This exit is called during volume level selection for all archive volumes in the input files data set. The user may use any and all data in the ARCHVOLS record provided to determine if the volume should be selected for XCOPY processing. The ARCHVOLS record passed to the user exit is a copy of the ARCHVOLS record used during the process and is discarded on return from the user exit. The Flag byte passed to the exit may contain a 'Y' or 'N' depending on whether the volume has been selected for processing via the selection criteria specified on the SELECT and XCOPY command lines. In this way the exit can know if the volume was to be selected and override processing if desired.

Table 5-93. Parameters Passed to XCPARCEX

Type	Size	Description	
INPUT	XL110	Input ARCHVOLS Record	
IN/OUT	CL1	Flag, C 'Y'	Select volume
		Flag, C 'N'	Do not select volume

XCPDSNEX — XCOPY Data Set Level Selection Exit

This module is called during XCOPY processing to allow selection processing based on criteria other than that available through SELECT and XCOPY parameters.

This exit is called for all data sets that exist on the Archive Volumes that have previously been selected for processing. The user may use any and all data in the ARCHVOLS and DSNINDEX records provided, to determine if the data set should be selected for XCOPY processing. The ARCHVOLS and DSNINDEX records passed to the user exit are copies of records used during the process and are discarded on return from the user exit. The Flag byte passed to the exit may contain a 'Y' or 'N' depending on whether the data set has been selected for processing via the selection criteria specified on the SELECT and XCOPY command lines. In this way the exit can know if the volume was to be selected and override processing if desired.

Table 5-94. Parameters Passed to XCPDSNEX

Type	Size	Description	
INPUT	XL110	Input ARCHVOLS Record	
INPUT	XL256	Input DSNINDEX Record	
IN/OUT	CL1	Flag, C 'Y'	Select data set
		Flag, C 'N'	Do not select data set

Chapter 6. Message Formatting

Module ADSUT028 is available to allow user-written programs (for example, a user exit) to write messages to the same //MSGPRINT dd statement used by SAMS:Disk. The user program merely formats the message text as desired into a 115-character print line (with trailing blanks for messages shorter than 115 characters). Module ADSUT028 is then called, passing the name of the user's program, a message number to be assigned to the message, and the message text to be printed.

The module must be invoked through the SAMS:Disk module intercept interface (MI). The example shows how to do this. To use this interface, the SAMS:Disk load library must be available when you link-edit your program, as documented beginning on page 353 in this manual.

The module and its interface are as follows:

Module Name — ADSUT028

Module function - Print message to MSGPRINT dd statement:

Table 6-1. Parameters Available to Module ADSUT028

Parm #	Type	Size	Description
1	Input	CL8	Name of program issuing the message
2	Input	CL3	Message number to be used (Will be prefixed with "U")
3	Input	CL115	Message text

Example Use of ADSUT028

Assembler language call -

```
CALL MI, (DMSPGM,USERPGM,MSGNBR,MSGTEXT),VL
```

Assembler constants for call -

```
DMSPGM    DC    CL8'ADSUT028'
USERPGM    DC    CL8'xxxxxxxx'
MSGNBR     DC    CL3'001'
MSGTEXT    DC    CL115'message text'
```

Message Formatting

All printed messages are prefixed with a 13-character identifier, consisting of the 8-character name of the module issuing the message, an asterisk or a space, and the 4-character message number.

Some of the messages issued by SAMS:Disk modules are constructed by a parameter-driven module that uses a combination of passed parameters and values in an appropriate entry in the DMSMSGES member of the parmlib data set. Prior to Release 7.0, the module called to format these messages was ADSDM190. In Release 7.0, all calls to DM190 were changed to call ADSDM590 instead, using a slightly different parameter structure.

If you have written programs to call DM190, your messages will now be prefixed with USRD190*nnnn (where nnnn is your message number). An additional message will then be issued as follows: ADSDM190 9999 IF MI CONTROL STACK IS ACTIVE, MODULE ISSUING PRIOR MESSAGE WAS mmmmmmmmm. To eliminate these extra messages you must change your programs to call DM590 instead of DM190, and insert the 8-character name of your module as the first parameter being passed to DM590. For example, assume your program named USRPGM03 has the following call:

```
CALL MI, (DM190,MSG5021,PARM1,PARM2),VL
```

It should be changed to:

```
CALL MI, (DM590,USRPGM03,MSG5021,PARM1,PARM2),VL
```

with the following definitions needed as well:

```
DM590 DC CL8'ADSDM590'
USRPGM03 DC CL8'USRPGM03'
```

Each entry in the DMSMSGES member of the parmlib data set consists of a message number, post-message construction action code, and a series of one or more field and literal specifications. The format of each entry, which is variable in length, is described below.

Table 6-2. Format of the DMSMSGES Member of PARMLIB

Colums	Description		
1-4	Message Number - A four-digit zoned decimal number that uniquely identifies the particular message		
5	Action Code - A one-character designation of action to be taken after the message is constructed and printed. It may be one of the following:	C	Continue by returning to the calling module
		T	Terminate Without A Dump and user code 16
		D	Terminate With A Dump and user code 20

Columns	Description		
6 - n	Field / Literal - One or more variable-length entries may appear, with the first character of each item describing the type of data to be inserted next in the message:	C	Designates a CHARACTER field of a length specified by the next one or two digits. The next parameter points to the value to be moved to the message.
		D	Indicates that a DATE in the (ydd) discontinuous binary format is to be processed as the next parameter
		X	Specifies a HEXADECIMAL field of length specified by the next digit. The value to be unpacked and translated is at the location pointed to by the next parameter. It may be 1 to 4 bytes in length.
		F	Specifies a FULLWORD field. The value to be unpacked and translated is at the location pointed to by the next parameter.
		H	Specifies a HALFWORD field. The value to be unpacked and translated is at the location pointed to by the next parameter.
		/	Indicates that a LITERAL begins with the next character and continues up to a terminating slash or the end of the member entry

Whenever an error is encountered by the message processing module, it will print an error message consisting of the prefix (ADSDM590) followed by appropriate error text. It will then print the completed portion of the message it was directed to construct. Depending on the nature of the error, this may be only the identification of the calling module.

Chapter 7. Problem Analysis/Aids

Sterling Software, Inc. takes pride in providing leading edge products using state of the art software technology. Providing high quality technical support is of equal importance. In an effort to maintain this standard for an increasing customer base, we request your assistance in performing as many of the following diagnostic steps as are relevant prior to contacting the Technical Support Center.

- ☐ Read the documentation carefully.
- ☐ Reconstruct the events and write them down.
- ☐ Run some simple test cases.
- ☐ If the job abended, find the name of the data set that SAMS:Disk was processing at the end of the Module Control Stack (refer to topic "*Module Intercept Control Stack*" on page 324 for details). If you were running an implicit function (that is, DMS), try an explicit run (that is, ARCHIVE) against the offending data set. If this abends, bypass the data set in your implicit run until you find out what is special about the offending data set.

If a problem arises that requires technical programming or analysis support, contact us at the following telephone number.

(800) 889-0226

We recommend that you not ask for a specific support representative unless you are already working with that individual on your particular problem.

When You Call In

When you call in, you will need to provide the following information:

- ☐ Your organization name: _____.
- ☐ Your name: _____.
- ☐ A phone number where you can be reached: (____) ____-____ x____.
- ☐ The CASE number or PI tracking system number _____, if one was assigned to the problem previously by a support rep.

For new problems, you will need to provide:

- ☐ The release of SAMS:Disk that you are running: _____.

- ☐ The machine type on which you are running (that is, IBM 3090, Siemens):
_____.
- ☐ The operating system and release on which you are running (that is, MVS 5.2):
_____.
- ☐ If the request is for a SAMS:Disk Open SVC zap, the PTF UZ number that you are running under: _____.
- ☐ The function(s) of SAMS:Disk that you are running: _____.
- ☐ If this is a security-related problem, the security package type, release, options installed, and the authority level of the user: _____.
- ☐ Can you recreate the problem? _____.
- ☐ Has the problem occurred before? _____.
- ☐ Has the function ever worked correctly? _____.
For how long? _____.
Under what conditions?
_____.
- ☐ Any changes made to your system between the time that SAMS:Disk worked and it didn't work: _____.
- ☐ The exact sequence of events that led to the error:

_____.
- ☐ The JCL from all jobs involved.
- ☐ The module name, message numbers and complete text of any SAMS:Disk messages that were issued.
- ☐ If you are running auto-restore, examine the message output from the DMSAR started task. If you are not getting any message output, set the //MSGPRINT DD SYSOUT= statement in the DMSAR proc to be printed or routed to your favorite spooling package. Debugging of auto-restore cannot be accomplished without the MSGPRINT output.

- ☐ The message numbers and complete text of any system messages that were issued (that is, from the JES log).
- ☐ The console log from around the time of the error, if it contains messages that do not appear on the JES log. For instance, I/O errors or bad data sets may result in KAA000I messages, and PFD 806 abends may result in other KAAxxxxi messages that will appear on the console log and not on the JES log.
- ☐ If there is a dump, the abend code and module intercept control stack information. Locate the SAMS:Disk control stack in user subpool zero, and identify the failing module. See “*Module Interface Control Stack*” on page 324 for details.

Note: Because SAMS:Disk uses GETMAIN areas for much of its processing, an ABEND-AID dump is almost never sufficient to troubleshoot problems in SAMS:Disk. Therefore, SAMS:Disk uses the ABNLDUMP dd statement to get a complete SYSUDUMP in addition to an ABEND-AID dump.

SVC dumps have an KAA911E message on the JES log, or the second word of the PSW in the dump points to a location not contained in the SYSUDUMP. If you have such an SVC dump, print it using the LPAMAP, LOGDATA, and PRINT CURRENT commands of the program AMDPRDMP. Also use the LISTLPA command of the program AMBLIST to get the locations of the system modules (AE users must follow other procedures.)

If you encounter a problem while running SAMS:Disk Stand-Alone Restore (DMSSAR), vary a printer online and retry the restore. If you go into a wait state, record the contents of the PSW. If a dump is produced, let the entire dump print. If you need to contact the Technical Support Center for assistance, please try to call from a phone near the console you are using for the restore because the support representative may need you to display certain portions of memory or other information at the console.

- ☐ A list of all active system parameters. You can specify sysparm SYSPARMS with a value of S to get this list.
- ☐ A list of all user exits that you have installed in SAMS:Disk.
- ☐ A list of all user exits in other products at your site, if they are related to the SAMS:Disk function that you are trying to run (that is, RACF user exits for problems with the SAMS:Disk/RACF interface).
- ☐ Results from tests that you have run.
- ☐ Any other related output.
- ☐ Any other information that may help to solve the problem.

☐ What exactly do you want the Technical Support Center to do? _____

_____.

Incomplete information may delay or prevent solutions to problems. If we do not receive a response within four weeks for a request for more information, we will consider the problem closed. If it is not possible to provide the information within four weeks, let us know so we can keep the problem open.

When you receive a fix to a problem, please confirm, good or bad, the results of the fix to the support rep. If we do not receive a response within two weeks, we will consider the problem closed.

This policy is to help us give you better service by responding to your problems in a timely manner. If you have any suggestions for changes to this policy, please contact the Technical Support Center. We always welcome suggestions to provide better service.

Sending Information to the Technical Support Center

In some cases the SAMS:Disk Technical Support Center representatives may request that you send information regarding a problem directly to them. Please document the problem completely before sending it, including:

- ☐ Your organization name and address.
- ☐ Your name.
- ☐ A telephone number where you can be reached.
- ☐ The PI problem tracking system number assigned by the support rep. Write this PI number on ALL documents and on the OUTSIDE of the package.
- ☐ Other information from the list above. The list itself may be photocopied and included with the package.
- ☐ The importance of the problem, so that we may assign it a priority.

If you have to send a dump in to the Technical Support Center, please provide the dump on paper (or if it is large, on tape). Dumps on microfiche will be accepted only if your installation cannot provide it on paper or on tape. Organizations that cannot ship dumps or other listings due to security restrictions may or may not receive assistance over the phone.

You may fax small numbers of pages to our facsimile machine at (916) 852-8601. If you cannot use the fax, send all information to the following address:

*Sterling Software, Inc.
Storage Management Division
11050 White Rock Road, Suite 100
Rancho Cordova, CA. 95670-6095*

Diagnostic Tools

SAMS:Disk is composed of numerous callable modules, each of which performs a separate unique function. SAMS:Disk components are driven by executing a control module which, in turn, calls all the lower-level modules necessary to complete the desired task.

Module Intercept Control Stack

Calls between all modules are done through a common interface routine, MI. This routine not only provides an interface between modules, but also provides the base for detailed debugging facilities. Three major functions are provided by the MI interface.

For each call, MI is passed the name of an executable load module. MI first determines if the module is already in storage. If it is, linkage is established, and control is passed to it. If it is not in storage, MI issues a load for the module and then passes control to it. Upon return to MI from the subroutine, MI returns control to the original caller.

During this process of interfacing between driver and subroutine modules, MI also keeps track of all modules currently active. Information about all active modules is kept in a table referred to as the module intercept control stack. The control stack is not maintained when MI fast path is used, which will be discussed shortly.

At the time of the first call through the interface, a control block is dynamically constructed in subpool zero. To facilitate finding it in a memory dump and making use of its contents, it is allocated on a 32-byte boundary and is enclosed with graphic headings.

This control area includes a complete description of the active portion of the hierarchic control structure of the executing program. It collects in a structured format information that is scattered through the user partition or region.

Finding the Module Intercept Control Stack

The module intercept control stack may be found in any SAMS:Diskabend SY-SUDUMP listing. Instructions for getting a dump during abends of PFD functions are supplied in the SAMS:Disk installation library in the member SPFTRACE. Use the following steps to find the module intercept control stack:

1. Find the INDEX for the dump. It is the last page of the dump.
2. Go to the page for entry "USER SUBPOOLS".
3. Within the next few pages you should see the STACK in the right-hand portion of the listing. It will look like the example shown.

Example of Module Intercept Control Stack

The following example shows how the module intercept control stack will appear in SAMS:Disk SYSUDUMP listings:

```

00000 00000000 *.....*
00000 00000000 *.....*
00000 00000000 *.....*
0E2E3 C1C3D240 * MODULE INTERCEPT CONTROL STACK *
5D740 40D7D340 *LVL MODULE   RS  SA  RT  EP  PL *
00000 00000000 * 0 EXTERNAL .....*
95128 00000000 * 1 ADSDMI048 .....*
A4BA0 0009565C * 2 ADSDM180 .....*
B6498 000A5880 * 3.ASDM181 .....*
B27A0 000B6A00 * 4 ADSDM008 .....*
A1678 000A4014 * 5 .....*
          .      .      .
          .      .      .
          .      .      .
          .      .      .
00000 00000000 *49 .....*
44040 40404040 *      0000033 CALLS ISSUED      *
7E7E7 E7E7E7E7 *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
04040 40404040 * .....*
04040 40404040 *      DMS DIAGNOSTIC AREA      *
04040 40404040 * .....*
440C9 E2406040 * DSNNAME OR F1DSCB AT ABEND IS . *
04040 40404040 *ADS.DMS.TEST.DSN01 .....*
04040 40404040 * .....*
04040 40404040 * .....*
440C9 E2406040 * VOLUME OR F4DSCB AT ABEND IS . *
04040 40404040 *WORK01 .....*
04040 40404040 * .....*
04040 40404040 * .....*
5C140 40404040 *      END OF DIAGNOSTIC AREA    *
7E7E7 E7E7E7E7 *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
00000 00000000 *.....*
00000 00000000 *.....*

```

Figure 7-1. Module Intercept Control Stack Listing

A brief description of each of the fields comprising a stack entry is described below:

Table 7-1. Module Intercept Control Stack Fields

Field Name	Description
Control Level (lvl)	The hierarchic level of function with the operating system being considered level 0. Up to 26 control levels can be accommodated
Module Name (module)	The eight-character name of the module most recently invoked at that level in the stack. A flag of hex FF will appear in front of the module name that is active at the time of the dump. This will print as a period on the graphic side of the dump listing.

Field Name	Description
Save Area (sa)	The second fullword following module name contains the address of the SAVEAREA of that module. This field is established at the time the module calls a lower-level module by way of the MTF INTERFACE. Until the first call, it is zero.
Return Address (rt)	The third fullword following the module name contains the return address from the most recent call to a subordinate module via the MTF INTERFACE. As with the save area pointer, it will be zero until the first call.
Entry Point (ep)	The fourth fullword following the module name is the address of the entry point for the module. It is placed there by the MTF INTERFACE immediately before giving control to the module. This value can be subtracted from the return address to determine the displacement of the most recent call made by a module (CLIST).
Parameter List (pl)	The fifth fullword following the module name, and the last value on each line of the dump listing, is the address of the parameter list passed to the module. It will point to the address of the parameter following the name of the module being invoked.

At the bottom of the STACK is the NUMBER OF CALLS issued prior to abnormal termination. It can be used in conjunction with the Module Trace Facility to obtain a trace of module names and parameter values for a reasonable number of calls prior to the failure.

MI Fast Path

Introduced in Release 7.0 was a streamlined path through MI that we call “MI fast path”. As the name implies, the purpose of this facility is to reduce the overhead associated with tracing and control stack maintenance. Although this can improve system performance up to 30 percent (depending on the function being executed), it also eliminates the debugging capabilities of MI. The fast path is keyed off the dd statements supplied in the JCL. If any dd statements start with “MTF”, the slow path through MI will be executed, which functions exactly as versions prior to Release 7.0. However, if no “MTF” dd statement is found, the fast path is taken.

How does fast path work, and how does it affect debugging? Basically, the first call that any module issues to another goes through the standard MI interface. Prior to passing control to the called module, however, MI replaces the VCON to MI with an address to an offset inside a branching table. This causes all subsequent calls to be routed directly to the called module (with four intervening instructions) without MI seeing the call. This causes the control stack to be maintained only on the first

call from one module to another, which is useless for debugging purposes. We therefore overlay the control stack heading line (LVL MODULE RS, etc.) with a literal “** CONTROL STACK UNRELIABLE **”. This also causes the “number of calls issued” value to be considerably less than the actual value.

There are occasions where this path cannot be used to pass program control. The primary one to be aware of is when the called module name can change dynamically throughout the course of execution, as in the following example:

```
IF (CLI,OPTION,EQ,C'1')
    MVC MODNAME,=CL8'ADSDM415'
ELSE
    MVC MODNAME,=CL8'ADSUT208'
ENDIF
#CALL MODNAME,(PARM1,PARM2),VL
```

This causes a problem in fast path processing because the VCON that points to MI will be replaced with an address that points directly to the module being called. Unfortunately, this will cause all of the following calls through this code to be passed to the first module that was called, even if the module name changes. This can be circumvented in two ways. The first is to include any dd statement that begins with MTF (such as MTFDUMMY). This will cause SAMS:Disk to use the slow path through MI for all program transfers. Although this will achieve the desired results, you will also be giving up the performance improvements in fast path.

This leads us to the second — and preferred — alternative of changing the linkage of MI to MI2. This tells the module intercept facility to not replace the VCON to MI for this particular call, but to go ahead and use fast path in any other calls (assuming no MTF dd statements were specified). So the updated code for the call above would look like this:

```
#LINK MODNAME,(PARM1,PARM2),VL
```

The other area of concern may be if you are using the MI facility to transfer calls between your own modules. If you have any assembler routines such as the one above, they should be changed to call MI2. Also, if you are using macros that have a hard-coded “=V(MI)” to transfer control, these will quite possibly cause a problem.

Another area to watch out for is if you are using MI to transfer control between programs written in a high-level language, such as COBOL or even PL/1. The compilers for these languages generally produce only one VCON to MI, which can lead to the same problem as that described above where the module name in an assembler program can change dynamically. In these languages you should always transfer control through MI2, although specific code was put in the module intercept facility to detect calls originating from COBOL programs so that the address of MI would not be replaced.

DMS Diagnostic Area

Immediately below the module intercept control stack in SAMS:Disk dumps is the DMS diagnostic area. This area contains a data set name and volume serial pertaining to the data set being processed at the time SAMS:Disk abended. This data is maintained whether or not MI fast path being used.

Following the literal “DSNAME OR F1DSCB AT ABEND IS - ” is the name of the data set being processed. Following the literal “VOLUME OR F4DSCB AT ABEND IS - ” is the volume serial of the volume on which the data set resides.

Turn to page [324](#) for an example of the SAMS:Disk diagnostic area as it will appear in SAMS:Disk dumps.

Module Intercept Tracing

The third function performed by MI is the ability to generate a trace of all calls and returns of control between driver and subroutine modules. This facility is normally turned off during SAMS:Disk execution. It may be initiated if it is necessary to trace the interactions between modules to determine the cause of problems or errors.

This feature is controlled by providing special JCL dd statements. When MI detects these dd statements, all code necessary to perform the trace is invoked. This code is in the form of loadable modules and does not incur extra overhead if not used. Use of the trace facility requires roughly 30K extra memory.

The dd statements needed to invoke the MI trace facility are:

Table 7-2. MI Trace Facility DD Statements

DD	Description
MTFIN	An input stream (DD *) containing a single trace control statement (described later)
MTFINOPT	A second input stream. Control statements may be included to confine the trace to modules specified by the user. Omitting this dd causes all modules to be traced.
MTFINPRT	SYSOUT class that will contain images of control statements supplied in the MTFINOPT dd stream. This statement is required only if the MTFINOPT dd is also supplied.
MTFPRINT	SYSOUT class to which trace output should be directed
MTFPRT	SYSOUT class to which trace control statements should be directed
MTFTABLE	This dd refers to a table used by the MI trace facility to format the trace output. Sample JCL in member MTFTJCL of the SAMS:Disk installation library is supplied to create the table. Use of this table creates a formatted trace with titles and parameter names.

Sample JCL to trace batch runs is supplied in your SAMS.DISK.INSTALL member TRACEDD. A CLIST to allocate JCL to trace PFD functions is supplied in member SPFTRACE. Both provide the following JCL to activate a trace:

```
//MTFPRT DD SYSOUT=*
//MTFPRINT DD SYSOUT=*
//MTFINPRT DD SYSOUT=*
//MTFIN DD *
0000000/05000/00NNFB
Use this control card when using the MTFTABLE dd statement
0000000/05000/03YNVB
//MTFTABLE DD DISP=SHR,DSN=SAMS.DISK.Rnn.MTFTABLE
//MTFINOPT DD *
INCLUDE
MBR=ADS.....
```

Figure 7-2. JCL to Activate Module Intercept Tracing

MI Trace Control Statement Format

The following section describes the control statement image to be included in the MTFIN dd control statement stream. When this dd and control statement are omitted from the SAMS:Disk JCL stream, no processing will be performed. However, the MI control stack will be maintained in either case.

Table 7-3. Format of the MI Trace Control Statement

Cols	Description
1-7	Call Skip Count: number of calls to ignore prior to activation of the trace — Default is zero
9-13	Call Trace Count: number of calls to be listed during execution; the trace will deactivate after this count is exhausted — Default is 100
15-16	Fixed Format Parameter Length: byte length of parameter values to be printed in hex and graphic format when the display option is “f” — Default is 4. Special values in this field cause parameter trace routines to attempt to automatically determine the length of each parameter to be traced. Any other value entered causes that specific number of characters of each parameter to be traced.
	00 Attempt to automatically determine the length of each parm
	01 Attempt to automatically determine the length of each parameter and dump all format-1 DSCB areas as 140 chars.
	02 Same as 01 except it dumps all format-4 DSCB areas also
	03 Attempt to determine the length of each parameter but set a minimum trace length of 16 characters. Also, dump all format-1 DSCB areas as a full 140 characters.

Cols	Description		
17	Module Title Display Option:	Y	Produce titles. If module title cards are supplied, the Default is Y
		N	no titles
18	Module Selectivity Option:	Y	Indicates if the tracing is to be confined to those modules that have been defined in the attribute data set. Note: The trace count is not confined to selected modules.
		N	The Default is No confinement
19	Module Parameter Display Option:	F	Designates fixed format
		V	Designates variable format. If V is specified and either no parameter definitions are supplied or a discrepancy exists between the actual number of parameters and the definition count, the default will apply.
		N	The Default is that No values are to be listed
20	Entry / Exit Display Option:	X	Indicates that parameter values are to be displayed only at exit time
		B	The Default is to display parameter values at both entry and exit time

MI Trace Options Control Statement

In addition to the trace control statement supplied in the MTFIN dd input stream, the user may also supply trace module selection criteria through a different dd statement control stream. This allows the user to considerably reduce the amount of trace output that will be generated. Control statements are supplied through the MTFINOPT dd statement. The dd statement and control statements are optional.

All control statement data for these statements is freeform format. Only one option may be specified per statement image. Option name may begin in any column. The following control statements are supported:

Table 7-4. MI Trace Options and Definitions

Option	Definition
INCLUDE/ EXCLUDE	Either the word INCLUDE or EXCLUDE may be specified in a statement image to indicate that values specified on subsequent control statements are to be traced (include) or not traced (exclude). Only one include or exclude statement may be present in the input stream. This statement is REQUIRED.
MBR=	This statement allows the specification of a single member to be included or excluded in the trace output. More than one MBR= statement may be present to allow the selection of more than one member. If followed by “,I”, only calls IN are traced (“,O” for calls OUT, “,B” for both).
PREFIX=	This statement allows the specification of a member name prefix that will be used to validate members to be excluded or included in trace output. More than one may be supplied.
PATH=	This statement allows the user to specify a path of modules to be included or excluded from trace output. The member specified and all of its subordinate modules will be used for trace member validation.

MI Trace Output

Whenever the appropriate control statement specifications are satisfied, a TRACE LISTING is produced. In addition to the date and time, the following information is provided for each call and return.

Table 7-5. MI Trace Output Explanation

Item	Definition		
CTL LVL	Hierarchic control level of the module receiving control. The user control module is always at level 1.		
ACTIVE MODULE	Member name of the module receiving control by either a call or return. This column provides a chronological list of control location.		
XFR	Type of control transfers	C	call
		R	return
CALRET MODULE	The member name of the calling/returning module that is yielding control to the active module.		
PAR CNT	The number of parameters being passed by the calling module or being returned by the returning module.		
MODULE TITLES/ PARAMETER VALUES	Descriptive titles of up to 50 characters are placed on this line for the two respective module names. Also, the parameter values are listed on succeeding lines with or without descriptive names and variable format.		

When the variable format parameter display is active, trailing blanks will be truncated from parameter values on the listing.

Archive Tape Analysis Utility

This utility is provided to aid in determining the validity of an archive tape volume. In the event that an archive tape volume becomes damaged, this utility can be used to help determine the extent of the problem. It will indicate what data sets are still valid and which ones have been lost or damaged. This information can then be used to determine if data sets can be recovered from the volume and, if so, which ones.

This utility operates independently from other functions. To perform its function it simply:

- Scans the archive tape volume
- Retrieves information from the files data set that it finds in both the archive tape and the files data set
- Compares the BLKID for each data set (if available) and indicates any discrepancies by:
 - placing a *D at the end of the printed line that contained the discrepancy
 - placing an * at the end of the printed line if the values compared equally
- Interprets its contents
- Produces a printed report

Executing the Utility

Use the following JCL as an example to execute the utility:

```
//STEP1    EXEC    PGM=ADSMI002,PARM='ADSDM455',REGION=1024K
//STEPLIB  DD      DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD      DUMMY
//FILES    DD      DSN=SAMS.DISK.FILES,DISP=SHR
//PARMLIB  DD      DSN=SAMS.DISK.PARMLIB,DISP=SHR
//CMDPRINT DD      SYSOUT=A  -- commands
//MSGPRINT DD      SYSOUT=A  -- error messages
//SYSPRINT DD      SYSOUT=A  -- report
//SYSUDUMP DD      SYSOUT=A
//SYSUT1   DD      DSN=the.archive.tape.data.set.name,
//          UNIT=TAPE,VOL=SER=vvvvvv,DISP=OLD
//SYSIN    DD      *
          ANALYZE ....
/*
```

Figure 7-3. JCL to Execute Archive Tape Analysis

Modifying the SYSUT1 DD Statement

The SYSUT1 dd statement refers to the archive tape volume to be scanned. If the archive tape volume is a multivolume data set, include all volumes in the JCL. You must provide the correct name of the “archive tape” on the SYSUT1 dd statement.

If the beginning portion of your SAMS:Disk tape has been accidentally overwritten, the operating system will no longer recognize the SAMS:Disk archives as the first standard label file on the tape. If you have this condition, an attempt can still be made to process any remaining SAMS:Disk archive data, by adding the following parameter to the SYSUT1 dd statement:

```
//          LABEL=( 5 ,BLP ) ,
```

This will cause the OPEN SVC to go past enough tape marks to get to the remaining portion (if any) of the SAMS:Disk archive data set. You may have to experiment with the number on this parameter, depending on the exact nature of the problem. If additional errors occur trying to read the tape (such as wrong length records), you can attempt to bypass these by adding the following DCB parameter to the same SYSUT1 dd statement:

```
//          DCB=(BLKSIZE=32760,DSORG=PS,RECFM=U) ,
```

If SAMS:Disk is still unable to read this tape, analysis and recovery of any remaining data cannot be done. If you have other tape copy/salvage utilities in your installation, you may be able to use them to create a new tape from the damaged one, from which SAMS:Disk can then analyze and recover as much data as possible.

Command Input

The utility accepts a command as input. It allows the user to specify parameters to control tape processing. Only ANALYZE is required. The command format and parameter descriptions follow:

```
ANALYZE COUNT=nnn ,ERRORMAX=nnn ,DUMPMAX=nnn ,EXITNAME=xxxxxxxxx ,
        FILENBR=nnn
```

COUNT=

Specify a 1- to 5-digit number to limit the maximum number of tape records to be scanned. It cannot be larger than 32,767. When this number of records has been read, processing will terminate. Due to error recovery, this number may be exceeded. It defaults to 9,999,999 records, causing the entire tape to be processed. This parameter is optional.

ERRORMAX=

Specify a number between 0 and 32767 indicating the maximum number of errors to be encountered before processing is terminated. It defaults to a limit of 10 errors before termination. This parameter is optional.

DUMPMAX=

As errors with data blocks are encountered, the data blocks are dumped to the MSGPRINT DD SYSOUT stream. Specify a 1- to 5-digit number to limit the number of data blocks to be dumped. When this number is reached, processing will continue, but no more blocks will be dumped. It defaults to a limit of 10 blocks. This parameter is optional.

EXITNAME=

Specify a 1- to 8-character user exit module name to be invoked. This optional parameter can be used to cause the utility to pass logical records from the archive tape to a user's processing routine. When specified, the user exit module will be invoked once for each archive tape logical record processed (logical record formats are described in this manual beginning on page 529 in this manual). Compressed records will be decompressed before being passed to the exit.

FILENBR=

Specify the 1- to 5-digit file number with which to start tape analysis. It defaults to the first file. This parameter is optional.

Sample Report

In addition to the printed report, error messages will be generated for all detected error conditions.

ARCHIVE TAPE VERIFICATION										PAGE	1
JUL 1, 1996											
ARCHIVE TAPE VOLUME = 200328, CREDIT = 07/01/1996, EXPDT = 00/00/0000, DSNAME = ISPDLM1.DMSN.C1996183.T131910											
ARCFLAG=00, ARCFLAG2=03, ARCHKCNT= 25, ARCDSCNT= 16, ARCBTCNT= 460,058											
FILE	DSN->ARVOL	FLAG1	FLAG2	BLKCO	SOURCE	ARC	ARC	STARTING	STARTING	ENDING	TOTAL
NBR.	-----	-----	-----	-----	VOLUME	DATE	TIME	BLK-PHYS	BLK-REC	BLOCK NO	BLOCKS
	-----	-----	-----	-----	BLKCT						PHYSICAL
	-----	-----	-----	-----							BLKNO
1	ISPDLM1.DMS.FILES				SSL807	07/01/1996	1319	1	1	1	1
	200328	08	00	0001	00000001						01000004
											01000004*
2	ISPDLM1.DMS.PARMLIB				SSL807	07/01/1996	1319	2	2	8	7
	200328	08	00	0007	00000007						01000005
											01000005*
3	ISPDLM1.DMSORCMD.D08NOV95.T100656				MVSS02	07/01/1996	1319	9	9	9	1
	200328	08	00	0001	00000001						0100000C
											0100000C*
4	ISPDLM1.DMSORCMD.D09NOV95.T104110				MVSS02	07/01/1996	1319	10	10	10	1
	200328	08	00	0001	00000001						0100000D
											0100000D*
5	ISPDLM1.DMSRRPTS.D08NOV95.T162717				SY9SY2	07/01/1996	1319	11	11	11	1
	200328	08	00	0001	00000001						0100000E
											0100000E*
6	ISPDLM1.D950504.T081521.CTLF.INPUT				SSL804	07/01/1996	1319	12	12	12	1
	200328	08	00	0001	00000001						0100000F
											0100000F*
7	ISPDLM1.D950508.T110808.CTLF.INPUT				SSL805	07/01/1996	1319	13	13	13	1
	200328	08	00	0001	00000001						01000010
											01000010*
8	ISPDLM1.ISPF.ISPPROF				SSL804	07/01/1996	1319	14	14	15	2
	200328	08	00	0002	00000002						01000011
											01000011*
9	ISPDLM1.ISPF.PROFILE				SSL802	07/01/1996	1319	16	16	16	1
	200328	08	00	0001	00000001						01000013
											01000013*
10	ISPDLM1.LOGON.CLIST				SSL805	07/01/1996	1319	17	17	17	1
	200328	08	00	0001	00000001						01000014
											01000014*
11	ISPDLM1.PROFILE.CLIST				SSL805	07/01/1996	1319	18	18	18	1
	200328	08	00	0001	00000001						01000015
											01000015*
12	ISPDLM1.SPFLOG1.LIST				SSL802	07/01/1996	1319	19	19	19	1
	200328	08	00	0001	00000001						01000016
											01000016*
13	ISPDLM1.SPFTEMP0.CNTL				SSL803	07/01/1996	1319	20	20	20	1
	200328	08	00	0001	00000001						01000017
											01000017*
14	ISPDLM1.SPFTEMP1.CNTL				SSL805	07/01/1996	1319	21	21	21	1
	200328	08	00	0001	00000001						01000018
											01000018*
15	ISPDLM1.TSOMAIL				SSL803	07/01/1996	1319	22	22	22	1
	200328	08	00	0001	00000001						01000019
											01000019*
16	ISPDLM1.UTIL.CNTL				SSL807	07/01/1996	1319	23	23	25	3
	200328	08	00	0003	00000003						0100001A
											0100001A*
ADSDM455 2110 ARCHIVE TAPE SCAN ENDED, TOTAL BLOCKS (25), TOTAL FILES(16), TOTAL LOGICAL RECORDS (597)											

Figure 7-4. Sample Archive Tape Verification Report

User Exit — Archive Analysis

A user exit may optionally be invoked to provide user programs access to archive tape data. This may be useful in recovering data. The user exit for logical records is passed the following parameters:

Table 7-6. Archive Analysis User Exit

Type	Size	Description	
INPUT	HW	Call Type	When the user exit is called with this halfword containing a value of 1, a logical record from the archive tape is being passed to the exit module.
			When the halfword contains a value of 2, an end-of-file (not end-of-volume) condition has been encountered on the input tape. No record is being passed. This gives the exit a chance to perform end-of-file processing.
INPUT	variable	Data Record	When parameter 1 contains a 1, this parameter contains a logical record from the archive tape. The contents should not be used in other cases. Do not use this parameter when parameter 1 is equal to 2.
OUTPUT	HW	Return Code	This parameter indicates if the user exit routine wants processing to continue passing more records or to terminate processing.
		H'1'	Continue Passing Records to the Exit
		H'2'	Terminate Processing, Close File and End

Archive Extract Utility

This utility extracts data records from SAMS:Disk archive volumes. It is normally used to recover data for a data set on damaged archive tapes. It scans the archive tape volume for the specified data set, and then copies the data records or blocks to a sequential data set. For sequential data sets, the output data set is the same format as before it was archived. It may be used directly by application programs. For other data set organizations, the extracted data must be processed by other utilities to reload and recreate the original data set.

This program actually functions as a user exit through the archive analysis utility. The program reviews logical records from the archives, and writes the selected ones

to an output sequential file. The user provides the output sequential file to be written to. It must contain proper DCB attributes for the data set to be extracted. Partitioned data sets cannot be recovered by this utility. Only one data set may be extracted for each execution of the utility.

This program places in the output file data that can most easily be processed by the user. In many cases the data is extracted into the exact format from which it was archived. In other cases the data will be input to a reload utility to actually recover the data into a usable format.

Also note that when SAMS:Disk encounters data sets with records longer than 32,760 bytes, those records will be split up into records of 32,760 bytes, and message 2961 issued to let you know this occurred.

Output from the Extract Utility

Following are descriptions of the contents of the output file for the different data set organizations processed:

DSORG=PS and DA (PHYSICAL SEQUENTIAL and DIRECT ACCESS)

The data blocks from these files are written sequentially in the same order that they were originally read from disk. The DCB attributes are taken from the archived data set's VTOC information that was stored in the archives. For a DA file being extracted, you should also specify DCB=DSORG=DA on the //SYSUT2 dd statement.

Sequential data sets and DA files with fixed-length records should be usable directly as extracted, although for DA files you must also ensure that all special attributes needed are assigned to the data set before being used by application programs.

However, DA files with variable or undefined-length records (RECFM=V or U) will probably not be usable directly as extracted. These data sets usually have special formatting requirements. All records can be extracted, but the formatting needs are not known by the extract function.

If the data set was originally in the unmovable format, care must be taken to ensure it is allocated properly on the DASD device. These data sets must be extracted to the same device type from which they were archived.

DSORG=IS and AM (INDEXED SEQUENTIAL and VSAM)

The logical records archived from the original data set are written to the sequential output file. The output file is written with the DCB attributes specified on the JCL statement for the output file.

To recover the data set, the extracted logical records must be processed by the KQCAMS utility. These utilities will use the logical records to re-create the data set. If the ISAM/VSAM records are out of sequence, sort the records on their appropriate key fields before using DQCAMS to load them.

Executing the Extract Utility

To recover a data set, the user must know the name of the data set and the volume serial of the archive tape volume on which it resides. The EXTRACT command is to be placed in the command input stream of the archive tape analysis utility. Place it immediately after the ANALYZE command. (Turn to page 332 for a discussion on how to specify and process the ANALYZE command.) Specify ADSDM460 as the value to the EXITNAME parameter in the ANALYZE command to cause the EXTRACT utility to be invoked.

EXTRACT command

```
EXTRACT DSNNAME=, TIME=
```

DSNAME=

This required parameter specifies the name assigned to the data set to be extracted.

TIME=

If there is a possibility that more than one data set resides on the archive tape with the data set name specified, specify this parameter to further qualify the data set. Specify the time the data set was archived. This value must be equal to the archive time displayed in the DSNINDEX record for the data set. Use the SAMS:Disk LISTD function to determine this information.

JCL Example

Use the following JCL as an example to execute the utility:

```
//STEP1 EXEC PGM=ADSMI002,PARM='ADSDM455'
//STEPLIB DD DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A -- commands
//MSGPRINT DD SYSOUT=A -- error messages
//PARMLIB DD DSN=SAMS.DISK.PARMLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A -- report
//SYSUDUMP DD SYSOUT=A
//SYSUT1 DD DSN=archive.tape.data.set.name,
// UNIT=TAPE,VOL=SER=vvvvvv,DISP=OLD
//SYSUT2 DD DSN=dms.test,DISP=(,CATLG),
// UNIT=DISK,VOL=SER=vvvvvv,
// SPACE=(TRK,(5,5))
//SYSIN DD *
ANALYZE EXITNAME=ADSDM460
EXTRACT DSNNAME=SAMS.DISK.TEST
/*
```

Figure 7-5. JCL to Execute the Extract Utility

Modifying the SYSUT1 and SYSUT2 DD Statements

The SYSUT1 dd statement refers to the archive tape volume to be scanned. If the archive tape volume is a multivolume data set, include all volumes in the JCL. You must provide the correct name of the “archive tape” on the SYSUT1 dd statement. If the beginning portion of the tape has been accidentally overwritten, turn to topic *"Modifying the SYSUT1 DD Statement"* on page 333 for additional notes.

The SYSUT2 dd statement refers to the output data set to contain the extracted data. Specify data set name, volume, and allocation parameters for the output data set. You may use any data set name for the output data set on the SYSUT2 dd statement. For DA files, add DCB=DSORG=DA to the dd statement. For ISAM and VSAM, supply

```
DCB=(DSORG=PS,LRECL=ll,BLKSIZE=bb,RECFM=rr)
```

where ll, bb, and rr are appropriate for the logical records being extracted.

Archive Fix Utility

This utility is provided to aid in correcting errors that exist within archive tape volumes. It consists of a single job step that accepts input transactions and, based on their contents, copies all or specific archive data set files from the input archive tape volume to a new output volume. As the copy process continues, errors that are detected on the archive tape are corrected.

Of course, errors cannot always be corrected without loss of data because all the original data may not be present or is not complete. In cases where the error is simply a bad file number, it can be corrected without data loss. If the error is a truncated or missing record, the rest of the block and any split record at the beginning of the next block will be bypassed. The file may or may not be able to be restored by normal SAMS:Disk restore processing. It is then the user's responsibility to inspect the restored data set for errors and correct them manually. When records of an archived PDS member are corrected, messages are printed to indicate the members that may contain data errors.

It accepts as input a SAMS:Disk archive tape volume. It does not use the files data set for processing the input tape, so the tape need not be currently defined to SAMS:Disk. The SYSUT1 dd statement points to the archive volume to be scanned.

The output volume is referred to by the ARCHIVE0 and ARCHIVEC dd statements. They are coded exactly as they are for normal archive executions (see the *DMS PROC* example on page 26 in the *User's Guide* for details).

The input command specifies what files are to be copied from the input volume to the output volume. It may optionally specify a file number at which to start and a maximum number of records to process.

A user exit may optionally be invoked to provide user programs access to archive tape data at logical record level (EXITNAME). This may be useful in recovering data.

Note 1: The fix utility uses the standard SAMS:Disk archive modules to create the new output data set. This means that all applicable system parameters are used. If data compression is active, the new archive data set will be in compressed format, whether or not the input was. Conversely, the input may be in compressed format. But if compression is not active for this utility, the output will be in decompressed (expanded) format.

Note 2: For the same reasons as stated in the note above, a record is added to the ARCHVOLS subfile of the files data set for the new volume(s) created. This archive volume record is not complete and must be deleted from the file after the job has completed. This can be done through the use of the IXMAINT JCL procedure and the PURGE VOLUME= command. This may be avoided by creating a special temporary files data set to be used by this utility.

Use the following JCL as an example to execute the utility:

```
//STEP1   EXEC  PGM=ADSMI002,PARM='ADSDM554'
//STEPLIB DD   DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD  DUMMY
//CMDPRINT DD  SYSOUT=A  -- commands
//MSGPRINT DD  SYSOUT=A  -- error messages
//SYSPRINT DD  SYSOUT=A  -- report
//SYSUDUMP DD  SYSOUT=A
//PARMLIB DD   DSN=SAMS.DISK.PARMLIB,DISP=SHR
//FILES    DD   DSN=SAMS.DISK.FILES,DISP=SHR
//ARCHIVEC DD  DUMMY
//ARCHIVE0 DD  DISP=(NEW,KEEP,DELETE),DSNAME=archvol.dsn,
//              DCB=(BLKSIZE=20000,DSORG=PS,RECFM=U),
//              LABEL=EXPDT=00000,UNIT=(TAPE,,DEFER)
//SYSUT1    DD  DSN=archive.tape.data.set.name,
//              UNIT=TAPE,VOL=SER=vvvvvv,DISP=OLD
//SYSIN     DD  *
              COPY .....
```

Figure 7-6. JCL to Execute Archive Fix Utility

Modifying the SYSUT1 DD Statement

The SYSUT1 dd statement refers to the archive tape volume to be scanned. If the archive tape volume is a multivolume data set, include all volumes in the JCL. You must provide the correct name of the “archive tape” on the SYSUT1 dd statement. If the beginning portion of the tape has been accidentally overwritten, turn to topic *“Modifying the SYSUT1 DD Statement”* on page [333](#) for additional notes.

The ARCHIVE0 and ARCHIVEC dd statements refer to the output archive volumes. When specifying an installation-defined DASD esoteric name for the UNIT= parameter, do not specify a data set name in the override JCL. Instead, use the POOLDEFS member in PARMLIB to specify a data set name for ARCNAME0.

Command Input

The utility accepts a command as input. It allows the user to specify parameters to control tape processing. The command format and parameter descriptions follow:

```
COPY COUNT=nnn, FILENBR=(nnnnn,nnnnn,...), ALL, EXITNAME=
```

COUNT=

If this simple parameter is supplied, all files on the input tape will be copied to the output file.

FILENBR=

Specify up to 100 tape file numbers to be copied from the input volume.

ALL

Specify a 1- to 5-digit number to indicate the maximum number of tape records to be scanned. When this number of records has been read, processing will terminate. Due to error processing, this number may be exceeded. Default is 32,760. Maximum is 32,767.

EXITNAME=

Specify a 1- to 8-character user exit module name to be invoked. This optional parameter can be used to cause the utility to pass logical records from the archive tape to a user's processing routine. When specified, the user exit module will be invoked once for each archive tape logical record processed (logical record formats are described in this manual beginning on page [529](#) in this manual). Compressed records will be decompressed before being passed to the exit.

Note: - COUNT, FILENBR, or ALL must be specified.

User Exit — Fix Utility: Parameters Passed

The user exit for logical records is passed the following parameters:

Table 7-7. Archive Fix Utility User Exit

Type	Size	Description	
INPUT	HW	Call Type	When the user exit is called with this halfword containing a value of 1, a logical record from the archive tape is being passed to the exit module.
			When the halfword contains a value of 2, an end-of-file (not end-of-volume) condition has been encountered on the input tape. No record is being passed. This gives the exit a chance to perform end-of-file processing.
INPUT	variable	Data Record	When parameter 1 contains a 1, this parameter contains a logical record from the archive tape. The contents should not be used in other cases. Do not use this parameter when parameter 1 is equal to 2.
OUTPUT	HW	Return Code	This parameter indicates if the user exit routine wants processing to continue passing more records or to terminate processing.
		H'1'	Continue Passing Records To The Exit
		H'2'	Terminate Processing, Close File And End

Repair Archvols (REPARCH) Utility

This utility (REPARCH) provides the ability to analyze and synchronize the archive volume (ARCHVOLS) records and the tape pool (DMSPools) records in the files data set. Its repair activities include:

- deleting corrupted ARCHVOLS records (broken chains)
- deleting unchained ARCHVOLS records
- synchronizing DMSPools record status with ARCHVOLS records

It also provides for a user exit that can be used to modify (convert) the device type field in the ARCHVOLS records.

Processing Considerations

For estimating purposes, allow approximately 30 seconds to process 1,000 ARCHVOLS records. For integrity reasons, this utility also has exclusive control of the files data set during processing, which should be considered along with the time estimate before running the utility.

Output

This utility generates a report listing all deleted ARCHVOLS records and the reason for deletion, and a total of the ARCHVOLS records that were processed and deleted.

JCL — REPARCH Utility

Use the following JCL to execute this utility. (A sample copy is in member REPARCH in the install library.)

```
//REPARCH EXEC PGM=ADSMI002,PARM=ADSDM636,REGION=3072K
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=*
//FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
//MSGPRINT DD SYSOUT=*
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
Specify command as described below:
```

Figure 7-7. JCL to Execute the REPARCH Utility

REPARCH Command and Parameters

REPARCH SIMULATE, PURGE=, DEVTEXT=, POOL=, TMS=

Table 7-8. REPARCH Command Parameter Descriptions

Parameter	Description			
SIMULATE	This parameter causes informational messages and report statistics to be issued, but no updates will be made.			
PURGE=	This parameter restricts the scope of ARCHVOL records that are to be examined and reconciled. Values that may be specified, and their meanings are:	VOLUME (the default)	All ARCHVOL records will be candidates for reconciliation.	
		NOPROCESS	No ARCHVOL records will be reconciled.	
DEVTEXT=	If you have a need to modify the device type field in one or more ARCHVOLS records, use this optional parameter to specify the name of the module you have written to perform the conversion. Parameters Passed To Exit:	Type	Size	Description
		INPUT	XL1	Simulation indicator X'01'
		INPUT	XL110	ARCHVOLS record image
		IN/OUT	XL4 4	byte device code
POOL=	This parameter restricts the scope of the SAMS:Disk tape pool records (in the DMSPools subfile) that are to be examined and reconciled. Values that may be specified, and their meanings are:	SYNCH (the default)	This parameter synchronizes the status of each record in the DMSPools subfile with the ARCHVOLS subfile. It returns to "scratch" status any pool records indicated as "in-use" but not having a matching ARCHVOLS record. It assigns "in-use" status to any DMSPools records that have a matching ARCHVOLS record.	
		FORMAT	Format tape pool based on the value specified for sysparm TPOOLSEQ.	
		NOPROCESS	No DMSPools records will be reconciled.	

Parameter	Description		
TMS=	This parameter sets the scope of CA1 record updates. This requires the TMSCTLEX user exit to be active.	EXPIRE (the default)	For those ARCHVOLS records that are deleted as a result of reconciliation, this option will result in CA1 expiring the volume. It requires that sysparm TMSCTLEX user exit be specified in sysparm TMSCTLEX.
		NOPROCESS	No CA1 updates will be performed.

Load Library Status Utility

This utility is a diagnostic facility intended for use by the Technical Support Center. It reports on the current status of the SAMS:Disk load library, indicating:

- Superzap fixes applied to modules
- Modules that have been modified since the base release
- Modules that are down-leveled from the base release

The report has two sections. The first lists the modules that have Superzaps. It also shows the zap date and IDRDATA information. The second lists the modules that are different from the base release. They may be different because of local modifications, maintenance, or library corruption. It also shows status (MODIFIED or DOWN LEVELED), current link date, and delivered link date.

JCL — Load Library Status Utility

Use the following JCL to execute this utility. (A sample copy is in member LOADLIB in the install library.)

```
//LOADLIB EXEC PGM=ADSMI002,PARM=ADSDM636,REGION=3072K
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOAD
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=*
//FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
//MSGPRINT DD SYSOUT=*
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
Specify command as described below:
```

Figure 7-8. JCL to Execute Loadlib Status Utility

Loadlib Command and Parameters

LOADLIB DSN=,VOLUME=,STATUS=,

DSN=

Supply the data set name of the SAMS:Disk load library in this required parameter.

VOLUME=

If the SAMS:Disk load library is not cataloged, supply its volume serial number with this optional parameter.

STATUS=

This parameter is used to specify the type of status operation to be done. The following option is available:

LIST (default)	Report on superzaps applied to modules, and modules not synchronized with the base release of SAMS:Disk.
-------------------	--

Track Dump Utility

The TRACK DUMP utility allows you to dump the data on a DASD track exactly as it appears on the track. It should be used in problem analysis if you question what data exists on a track.

This utility reads and prints the contents of a track from a DASD volume. The DASD volume can either be online or offline; however, a path must be available to the device (that is, the device must not be “boxed”).

For each track, this utility prints the home address, the capacity (R0) record and all data records on the track. The count information is also printed for the capacity and data records.

The utility accepts either a single absolute track address or a range of absolute track addresses. If the volume is offline, the channel unit address for the volume (CUU) must be supplied; if the volume is online, only the volser need be supplied.

Command parameters NOR and NOD are provided to control the amount of data actually printed.

Use the following JCL to execute this utility. (A sample copy is in member TRACKDMP in the install library.)

```
//TRACKDMP EXEC PGM=ADSMI002,PARM=ADSFH114,REGION=2048K
//STEPLIB DD DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD DUMMY
//SYSUDUMP DD SYSOUT=*
//CMDPRINT DD SYSOUT=*
//MSGPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//PARMLIB DD DSN=SAMS.DISK.PARMLIB,DISP=SHR
//SYSIN DD *
Specify command as described below:
```

Figure 7-9. JCL to Execute Trackdump Utility

TRACKDMP Command and Parameters

DUMP CCHH= , VOL= , NOD , NOR , CUU=

CCHH=

Specify an absolute track value, a list of values, a range, a list of ranges, or any combination. The track value is input as cccchhhh (cccc is the cylinder and hhhh is the head). These values are entered in hexadecimal format. Examples of acceptable input are:

```
CCHH=cccchhhh
CCHH=(cccchhhh,cccchhhh,...)
CCHH=cccchhhh-cccchhhh
CCHH=(cccchhhh-cccchhhh,cccchhhh-cccchhhh,cccchhhh)
```

Up to ten values may be provided in the list. A track range counts as one value.

VOL=

The volume serial for the DASD volume you are processing. This parameter is required only for an online volume.

CUU=

The channel unit address for the DASD volume you are processing. This parameter is required only if the volume is offline. If both the CUU= and the VOL= parameters are specified and the volume is online, the supplied volser is compared to the real volser and an error message is issued if they are not the same.

NOD

If this parameter is specified, the home address, R0 record and count fields for all data records on the track are printed. The data portion of the records are not printed.

NOR

If this parameter is specified, the home address and R0 record values are printed for each track. No portion of the data records are printed.

Diagnostic DD Statements

Certain SAMS:Disk functions will provide additional diagnostic data whenever they detect a specific dd statement being present. The Technical Support staff may request that you supply one or more of these dd statements when researching a reported problem.

//ARESDIAG DD SYSOUT=A

This dd statement causes the auto-restore function to interpret the control block that is used to initiate the started task; that is, the environment that invoked the auto-restore and the options that are in effect.

//MTFLIBDG DD SYSOUT=A

This dd statement causes the MI component to display information about each module being loaded; for example, the library in which it was found, the concatenation number, the SAMS:Disk time and date stamp, etc. It may be used in both batch and online functions.

Archive Tape Recovery Utility

The Archive Tape Recovery Utility is intended to perform the function of recovering SAMS:Disk archive tapes which have been damaged or partially overwritten. It will read a single archive tape as input and bypass I/O errors and optionally skip tapemarks to retrieve any archive data blocks which are still intact.

Each complete data set encountered will then be transferred to a new archive tape volume. This tape volume can then be used by the REBUILD utility to identify it to the files data set. The Archive Tape Recovery Utility is designed to recover only archive tapes. It does not use disk data sets as input or output.

The JCL to run this utility is:

```

/*****
/* TAPE RECOVERY UTILITY PROGRAM
/*****
//S1EXCP EXEC PGM=ADSMI002,PARM=ADSDM563
//STEPLIB DD DISP=SHR,DSN=your.dms.loadlib
//PARMLIB DD DISP=SHR,DSN=your.dms.parmlib
//CMDPRINT DD SYSOUT=*
//MSGPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1 DD DISP=SHR,VOL=SER=xxxxxx,LABEL=(1,BLP),
//          UNIT=TAPE,DSN=your.old.archive.tape.dsn
//SYSUT2 DD DISP=(NEW,KEEP),VOL=SER=(xxxxxx,xxxxxx),
//          LABEL=(1,SL),UNIT=(TAPE,,DEFER),
//          DCB=(DSORG=PS,RECFM=U,LRECL=0,BLKSIZE=32760),
//          DSN=your.new.archive.tape.dsn
//SYSIN DD *
TRECOVER ERRLIMIT=4050,EOFSTOP

```

Figure 7-10. JCL to Execute Tape Recovery Utility

Note: DDNAME SYSUT1 points to the input archive tape volume while SYSUT2 points to the output archive tape volume(s).

If the output tape cannot hold all information from the input tape, you may specify more than one output volume. If you specify only one tape volser and an additional tape is needed, this utility will issue a mount request for a scratch tape. If you create two output volumes, be aware that these tapes are not related to one another. That is, they are not part of a multi-volume group.

The command used to direct the execution is:

```
TRECOVER ERRLIMIT= EOFSTOP
```

ERRLIMIT=

This parameter indicates the maximum number of errors to allow before terminating the job. When this limit is reached, processing is stopped and the output tape is closed. The default value is 100.

EOFSTOP

This parameter causes processing to stop when double tape marks are encountered, indicating the end of volume (EOV). Specify this parameter if your archive tape has been damaged.

Do not specify the EOFSTOP parameter if your archive tape has been overwritten or you want to recover past the current end of volume. Processing will continue until the end of the input tape is reached. If any tape marks are encountered, they will be treated as embedded tape marks.

Expected return codes from this utility are as follows:

Table 7-9. Archive Tape Recovery Condition Codes

Code	Description
0	All data sets have been recovered successfully from the input tape
4	The entire input tape has been processed successfully. However, errors encountered have prevented the recovery of all data sets.
8	The error limit specified on the ERRLIMIT= parameter has been exceeded. Processing of the input tape has been halted; not all data sets have been recovered.
12	Either critical I/O errors have occurred which prevent further processing of the input tape, or OPEN errors for either the input or output tapes have occurred. Not all data sets have been recovered.
16	Command error or an invalid output device type has been specified.

If you receive a return code of less than 16, you still may have recovered some of the data sets to the output tape.

After you have run this utility, run the Archive Tape Analysis Utility to verify that the new archive tape is good. (Just because the Archive Tape Recovery Utility completes successfully does not guarantee the tape is a good archive tape.) If the Archive Tape Analysis Utility indicates the tape is good, use the tape as input to the REBUILD utility. Then run PURGE on the damaged archive tape.

Chapter 8. Modifying SAMS:Disk

We hope that the combination of system parameters and user exits provided will eliminate the need to modify any of the SAMS:Disk source modules. Doing so obviously adds to the difficulty of installing any updated version of the system, since the same or comparable changes will have to be made to the new system as well. Should a source change be needed, however, concerns unique to the assembly and link-edit of SAMS:Disk modules are presented here.

A common interface module (MI) has been statically linked with each SAMS:Disk module that calls any other module. A second interface module (MI2) is also available, and should be used in programs you write that call other SAMS:Disk programs. Turn to the topic “*MI Fast Path*” on page [326](#) for more information about usage of MI2 versus MI. Both are included in the SAMS:Disk load module library for linkage editor processing when assembling modules at your site.

The addressing mode and residency mode of the SAMS:Disk module that you have modified should remain unchanged. If you change the original program attributes, unexpected results could occur.

You are responsible for changes you make to SAMS:Disk. This applies to any change to a SAMS:Disk module, whether made through source code changes or zaps. If you change a SAMS:Disk module, you are also responsible for properly installing any fixes provided for that module. This means, in part, removing any usermods you have installed to the affected module.

We recommend you make changes as usermods if you install SAMS:Disk using SMP/E.

Assembly and Link-Edit of SAMS:Disk Modules

If changes are made to the SAMS:Disk source modules, and for user-written programs using the MI or MI2 interface, the following steps must be taken to obtain proper assembly and link-editing.

1. SAMS:Disk source code uses structured programming macros. The SAMS:Disk macro library must therefore be concatenated to the assemble macro library as defined on the `//ASM.SYSLIB` dd statement. The SAMS:Disk macro library should be placed first in the concatenation to avoid the possibility of duplicate macro names in conflict with your installation's macros.
2. Fujitsu High-Level Assembler (HLASM) must be used to assemble SAMS:Disk modules. We will try to allow source members in the SAMS:Disk.INSTALL library to be assembled with Assembler F.

However, other SAMS:Disk programs may have incompatibilities that will require Fujitsu's HLASM.

3. At link-edit time the parameters should specify LET, but not NCAL.
4. At link-edit time the SYSLIB dd statement must point to the SAMS:Disk load library (DSN=SAMS.DISK.LOAD).

The following assembly JCL is provided on the distribution tape in the INSTALL library. You will need to customize several DD statements: STEPLIB, SYSLIB, SYSPUNCH, SYSLIN, SYSIN, SYSLMOD.

```

/*****
/* SAMS DISK MODULES HIGH LEVEL COMPILE AND LINKEDIT MODULES
/*****
/* USE THIS JCL FOR SAMS DISK LOAD MODULE UPDATES.
/*
/* USE LKED PARMS (RENT|REUS|REFR).
/*****
//HLASM EXEC HLASMCL,REGION.ASM=3072K,
//          PARM.ASM=(OBJ,NODECK,'PC(UHEAD,GEN)',NORLD),
//          PARM.LKED='LIST,MAP', <--- REUS,REFR,RENT
//          REGION.LKED=3072K
//ASM.SYSLIB DD DISP=SHR,DSN=SAMS.DISK.RNNN.FMACLIB
//          DD DISP=SHR,DSN=SYS1.FMACLIB
//          DD DISP=SHR,DSN=SYS1.FMODGEN
//ASM.SYSIN DD DISP=SHR,DSN=SAMS.DISK.RNNN.SOURCE(MODNAME)
//ASM.SYSTERM DD SYSOUT=*
//LKED.SYSLMOD DD DISP=SHR,DSN=SAMS.DISK.RNNN.LOAD(MODNAME)
//LKED.SYSLIB DD DISP=SHR,DSN=SAMS.DISK.RNNN.ADMSLOAD

```

Figure 8-1. Assembly JCL: Member ASMDISK

The following link JCL is provided on the distribution tape in the INSTALL library. You will need to customize several DD statements: STEPLIB, SYSLIB, SYSPUNCH, SYSLIN, SYSIN, SYSLMOD.

```

/*****
/* LINK A LOAD MODULE
/*****
/*
/* INSTRUCTIONS:
/* REPLACE "CALL" WITH "RENT", "REUS" OR "REFR"
/*****
//LINK EXEC PGM=JQAL,
//          REGION=3072K,
//          PARM='XREF,LET,LIST,CALL'
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DISP=SHR,DSN=SAMS.DISK.RNNN.LOAD
//SYSLIB DD DISP=SHR,DSN=SAMS.DISK.RNNN.OBJECT
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,3)
//SYSLIN DD DISP=SHR,DSN=SAMS.DISK.RNNN.LINKCNTL(MODNAME)
//

```

Figure 8-2. Link JCL: Member R900LINK

Issuing Messages

Please turn to the *"Message Formatting"* section beginning on page [315](#) for a description of how to issue messages from a user program or exit module.

Relinking TSS and PFD Control Modules

Each TSS command processor is associated with a module of the same name in the SAMS:Disk source library. Control modules for the PFD functions are also in the source library. They and all of the subordinate (or called) modules are dynamically loaded by the MI interface as they are needed. As a result, each module supporting these online functions can be reassembled and linked into the SAMS:Disk load library independently.

If you write user exit routines that affect the TSS or PFD functions, they too can simply be assembled and linked independently into the SAMS:Disk library. MI will dynamically load and execute your exit routines the same as any other SAMS:Disk modules, as soon as you supply the sysparms to activate them.

Adding Non-SAMS:Disk Modules

It is also possible to modify the system without changing or replacing modules, in the sense that SAMS:Disk modules may be invoked by non-SAMS:Disk calling modules. Each module given control through the common interface brings into memory those subordinate modules not already loaded. In this manner any function can be initiated in any environment as long as the required dd statements are present.

The Module Interface (MI) Trace aids in testing and verifying changes made to the system. It makes available a printed display of all calls and returns made through the interface module. It also displays the first "n" characters of each parameter value on both module entry and exit.

Coding for an Interactive Environment

If SAMS:Disk modules are to be used for interactive applications, special programming constraints are often imposed. A good technical understanding of the constraints of the environment as well as the design and constraints of the SAMS:Disk modules is required. The MI control modules account for these. Programming in other environments may introduce unexpected results.

Module Call Interface

Because of its dynamic nature, SAMS:Disk installation modules may use the internal interfaces in the same manner as SAMS:Disk system modules. To dynamically transfer control to a SAMS:Disk module, the installation module must call via the MI interface with the name of the SAMS:Disk module to be invoked as the first parameter.

The use of the SAMS:Disk call interface is illustrated below with an example in assembly language, since it will normally be the language used to invoke the various modules; however, COBOL may also be used to accomplish the invocation. Please note, however, the additional concerns regarding “*MI Fast Path*” as documented on page 326 of this manual.

Example — Call Interface

A unique record may be retrieved from a file in the SAMS:Disk files data set by executing a call in the following manner. This is an example only. Turn to the topic “*File Handler Interface*” on page 30 for specific information on using the DM095 interface.

```

MVC DSNKEY,=CL44'PAY.MASTER'
* RETRIEVE THE DATA SET INDEX RECORD
CALL MI,(DM095,RU,FILENAME,DSNREC,RESULT),VL
CLC RESULT,=H'1'
BH NOTFOUND

*
DM095 DC CL8'ADSDM095' FILE HANDLER CONTROL MODULE
RU DC CL2'RU' SERVICE TYPE
FILENAME DC CL8'DSNINDEX' IN - PARM 1 (FILENAME)
DSNREC DC 0CL256 IN/OUT AREA
DSNKEY DC CL44' ' IN - PARM 2 (KEY)
DC CL212' ' OUT - INDEX DATA
RESULT DC H'0' OUT - RETURN CODE

```

Figure 8-3. Example of Call Interface

User Exits

Turn to the “*User Exits*” section beginning on page 213 for a description of the modifications available via exits. A sample of user exit code is also provided in that section.

Parameter Library Support - Common Parmlib Interface

All functions of SAMS:Disk require various parameters to guide their execution. These parameters are maintained in a partitioned data set referred to as “parmplib”. This is a “read-only” data set for all SAMS:Disk functions. Any changes to be made to the parameter library must be done with an online editor or some other traditional means of updating a PDS.

User-written modules may read entries from a parameter library member in the same manner as SAMS:Disk modules, by using the common parmplib interface module. The syntax rules for parmplib entries begin on page 543 in this manual. The following illustrates reading an entry from member=EXMTLIST that may contain a list of data set names that are exempt from processing.

```

MVC  MBRNAME,=CL10'EXMTLIST'
* Retrieve an entry from the parmplib member
CALL MI,(ADSTS048,RESULT,MBRNAME,ENTRY#,MAXLEN, X
        IOAREA),VL
CLC  RESULT,=H'1'
BH  NOTFOUND

*
ADSTS048 DC  CL8'ADSTS048'  parmplib interface module
RESULT   DS   H              result
MBRNAME  DC   CL10' '        member name to be read
ENTRY#   DC   H'0'          entry number - 0 retrieves 1st
*                               record, 1 retrieves 2nd, etc.
MAXLEN   DC   H'46'         length of I/O area
IOAREA   DS   0CL46         data from retrieved entry
LENGTH   DS   H              length of record + 2
RECORD   DS   CL44          actual record

```

Figure 8-4. Common Parmlib Interface Module Example

The parmplib common interface module has the following call structure.

```
CALL MI,(ADSTS048,RESULT,MBRNAME,ENTRY#,MAXLEN,IOAREA),VL
```

Table 8-1. Description of the MI CALL Structure

Name	Description
ADSTS048	Is always the eight-character name of the common interface module to the parmplib: ADSTS048 DC CL8'ADSTS048'
RESULT	The halfword return code set by the control module to reflect the conclusion of the requested service. 1 indicates record found, 2 indicates end of file, and 3 indicates an invalid member name.
MBRNAME	The name of the member in the parameter library in a ten-character field.

Name	Description
ENTRY#	The number of the entry within the member that is to be read. Note that all numbers are relative to zero. The entry number is incremented within the interface module. An entry number of zero will retrieve the first record and the returning entry number will be one. An entry number of seven will retrieve the eighth record and the returning entry number will be eight.
MAXLEN	A halfword containing the length of the I/O area.
I/OAREA	An area in which to place the entry. The data from the entry is actually placed at an offset of two bytes into this area, with the first two bytes serving as a length descriptor. The halfword length field will contain the true length of the specific entry just returned, plus two bytes for the descriptor itself.

Common Sysparms Interface

Besides the TS048 interface to data in any of the parmlib members, a special common interface (DM175) is available for accessing system parameters from member=SYSPARMS. User-written modules may read sysparm values in the same manner as SAMS:Disk modules by using the common interface. For more information on sysparms, including sysparm names reserved for your use, turn to the "Sysparms" section beginning on page 103 in this manual. The following illustrates reading the VSAMSUPP sysparm.

```

* GET VSAMSUPP SYSTEM PARAMETER
CALL MI,(DM175,PARMNAME,LEN,PARMVALU,DM175RC),VL

DM175    DC    CL8'ADSDM175'
DM175RC  DC    H'0'
LEN      DC    H'1'
PARMNAME DC    CL8'VSAMSUPP'    name of sysparm desired
PARMVALU DC    CL1' '           sysparm value returned here

```

Figure 8-5. Common Interface DM175 Example

The following table helps explain the fields within the Common Interface DM175.

Table 8-2. Field Descriptions within DM175

Field	Description	
DM175	Is always the eight-character name of the common interface module: DM175 DC CL8'ADSDM175'	
PARMNAME	Is the eight-character name of the sysparm whose value is to be returned.	
LEN	Is the halfword field specifying the length (number of bytes) of the output value being returned. It must correspond to the length defined for the PARMVALU field.	
PARMVALU	The area where the value of the sysparm will be returned. Its length must equal the maximum number of characters allowed for the parameter, and agree with the value specified for the LEN parameter.	
DM175RC	The halfword return code from module ADSDM175.	
	0	Indicates that the system parameter was specified and has been returned.
	4	Indicates that it was not specified and the default value has been returned.

System Parameter Validation - Parmedit Member

For editing purposes, each SAMS:Disk sysparm has a system default value and validation criteria associated with it. Default information indicates the value to be used for each sysparm if you have not specified your installation values in member SYSPARMS of the parmlib data set.

Validation criteria indicate valid values for each sysparm. They are used when SAMS:Disk loads the sysparm entries into memory. This is done whenever a SAMS:Disk function is executed in either batch or TSS processing. The value of each sysparm supplied in the SYSPARMS member is checked against the validation criteria specified in the PARMEDIT member. If an improper value is supplied, SAMS:Disk issues an appropriate message. If the sysparm in error is subsequently accessed, SAMS:Disk will abend.

One entry is maintained in the PARMEDIT member for each SAMS:Disk sysparm. The PARMEDIT member is supplied with the SAMS:Disk system as part of the distributed parmlib data set, and may change with each release.

Although the PARMEDIT member entries can be modified, normally they should not be.

Note: Changes to defaults should be specified in your SYSPARMS member, NOT by changing the PARMEDIT member. The only common need for an update to the PARMEDIT member is to allow or disallow the overriding of a parameter in the input stream. The PARMEDIT member is replaced with each new release of SAMS:Disk. The SYSPARMS member is user-defined, and not replaced with each release.

Each entry contains the following fields:

Table 8-3. Format of PARMEDIT Entries

Cols	Description
1 - 8	System parameter name
10 - 12	Minimum number of characters required for parameter value.
14 - 16	Maximum number of characters allowed for parameter value.
18 - 20	Number of characters to be returned to SAMS:Disk module requesting the value of the system parameter.

Cols	Description		
22 - 23	Validation criteria:	'Y'	The system parameter value requires a Y or N value indicating either "yes" or "no" for the specified option.
		'N'	
		'NM'	The data for this parameter is of a numeric format. The user must enter numeric data but may enter only significant digits left-justified after the system parameter name.
		' '	No data validation will be done on data entered for this parameter value.
25	Enter Y if this system parameter is to be marked obsolete. Blank indicates valid system parameter.		
26	Enter Y if this system parameter may be overridden through the SAMS:Disk SYSPARMS dd input stream. Enter N if the parameter may not be overridden.		
28 - 29	(reserved for future use)		
31 - 100	The default value to be returned if the system parameter is not supplied by the user.		

Adding User Functions to SAMS:Disk PFD

There are two ways to add your functions to the SAMS:Disk PFD interface. One is to create a static menu selection panel and the other is to add them to the SAMS:Disk PFD dynamic menu builder. Before you make any changes, let's examine the structure and purpose of the SPFMENU member of PARMLIB. See the example below.

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6
'SP123    ONLIRPT Y 01  REPORTS - DEFINE YOUR OWN, EXECUTE',
' ONLINE (OPTIONAL)'
'SP026 L  LDSINDX Y 02  LIST - ARCHIVE/BACKUP INDEX ENTRIES'
'SP026 D  DDSINDX Y 02  LIST - (OR DELETE) ARCHIVE/BACKUP ',
'INDEX ENTRIES'

```

Figure 8-6. Example of SPFMENU Member of Parmlib

Starting in column 2 are the last five characters of the program name (the first three characters of the program must be ADS and are not specified here).

In column 8 is the parameter to be passed to the program. It may be blank (no parameter) or a single character.

The field in columns 11-17 is matched to options specified for each user in SPFOPTNS member. For more information, turn to the topic *"Defining PFD and DSCL User Options"* on page 67 of the *Installation Guide*.

Column 19 is a reserved field and is always Y.

Column 21 identifies the section of the condensed menu to which the function belongs. There are five sections. Keep the numbers in column 21 in ascending order as no validation is performed and unreliable results may occur otherwise.

Beginning in column 25 is the descriptive heading for the function. Lines can be continued and must be enclosed in quotes.

To add entries to the dynamic menu builder, follow these steps.

1. Create a program name that begins with ADS for your function.
2. Enter that name (minus ADS) beginning in column 2.
3. Set the parameter value in column 8.
4. Select a seven-character code for your function and enter it beginning in column 11. Also, you must specify this code in SPFOPTNS member in PARMLIB for the appropriate users.
5. Write your descriptive heading beginning in column 25.

See examples below:

```
.....1.....2.....3.....4.....5.....6
'REST1 Z  ZRTNONE Y 03  RESTORE - FUNCTION 1 '
'BACKP    BACKUPX Y 04  BACKUP DATA SETS '
```

Figure 8-7. Sample Dynamic Menu Builder

Note: The files and parmlib data sets are allocated by the dynamic menu builder. You can free and reallocate them if needed but be sure the SPFMENU, SPFTEXT and SPFOPTNS members are in the allocated parmlib.

To create a static menu selection panel, follow these steps.

1. Select the functions you wish to use in your panel.
2. Invoke them using standard PFD select statements.

The following are examples from the above SPFMENU example.

```
0, 'PGM(ADSSP123) '
1, 'PGM(ADSSP026) PARM(L) '
2, 'PGM(ADSSP026) PARM(D) '
```

Modifying SAMS:Disk PFD Functions

When modifying SAMS:Disk PFD functions, whether they are JCL/PROCS, panels or parameters, it is best to start with the SAMS:Disk PFD dynamic menu builder. The menu builder gives you the name of the SAMS:Disk module that controls that function (see “*Adding User Functions to SAMS:Disk PFD*” on page [361](#) for a detailed explanation of the dynamic menu builder).

Within the module are panel names, skeleton JCL names, and required parameters. (To understand these, you must be able to read source code.) A few modules may make calls to other modules to display different panels (that is, PFD Online Reporting).

For example, if you wished to remove the DELETE action from the archive function, first examine the SPFMENU member in PARMLIB to determine all archive functions. The first module listed for archive is ADSSP001. If you browse the source for ADSSP001, you will find the panel names DMS001A, DMS001B and DMS001C, and skeleton JCL names DMS001A and DMS001B. The DELETE action has no required parameters so the delete action line can be removed.

If, on the other hand, you want to change the PROC names to be executed, it would be much easier to simply scan the PFDPROCS for those PROC names.

Chapter 9. DSCL Processing

During the development of the DSCL support, one of the foremost concerns of the project was to develop a product that is user-friendly. Four major areas needed to be addressed:

1. Provide validation of all input commands prior to processing
2. Issue meaningful error messages whenever an error is encountered
3. Provide the flexibility that the user requires to make DSCL a workable tool
4. Provide maximum efficiency in execution time by scanning any catalog only once, regardless of the order of input commands entered by the user.

In an attempt to deal with human factors in the use of the IMPLICIT VSAM support, all commands are validated prior to any action being performed. If any errors are detected, the validation of commands continues, but no action is taken after validation is completed. As each command is read in and validated, an appropriate control block is built (discussed later) containing the pertinent data to the command. As processing continues, the control blocks are linked together to form the processing sequence that will be followed if all validation tests are satisfactory.

The execution order of commands is not necessarily the same order as they are entered by the user — the program performs all of the actions required in a single catalog scan prior to scanning the next catalog/volume (if any). This ensures that a catalog/volume is scanned only once in any given job run, regardless of the number of actions specified by the user. This can be somewhat confusing to the user who might expect results in exactly the same order as which the commands were entered.

Control Block Structure

Six different control blocks are employed by DSCL processing. Five of the control blocks relate directly to the input commands entered by the user. The sixth is the JOB Control Block (JOB CB), which contains informational data regarding the status of the job run. The JOB CB is built during initialization processing in ADSST001, and is linked to all of the other control blocks in the system by the field xxxJOB@ (where xxx is the control block type, such as CAR or LOC). The JOB CB contains such information as the job return code and the highest DSN expiration date encountered during archival processing. It also contains the highest RETPD specified on all ARCHIVE and BACKUP commands.

The Locate Control Block (LOCCB) corresponds directly to the SCAN command and is responsible for keeping track of which volumes(s)/catalog(s) to scan. For each SCAN command entered, one LOCCB is built. An exception is made if a SCAN command names the same catalog/volume as a previous SCAN request, such that any subsequent commands are chained off the initial LOCCB for that catalog. The LOCCB also contains any password supplied by the user necessary to open the catalog.

The Command Activation Request Control Block (CARCB) is built for each action command entered (such as BACKUP, ARCHIVE, or DELETE). It is linked directly under the LOCCB and is pointed to by the field LOCDOWN in the LOCCB. This control block informs the program which function to perform on a given catalog in the LOCCB. Multiple actions can be taken on a single catalog/volume scan, which is accomplished by linking multiple CARCBs to each other under a LOCCB. The link is provided by the CARNEXT field in the CARCB. The CARCB contains such information as the action type (DELETE, ARCHIVE or BACKUP), a call list that is executed by ADSST201, and command parameter information.

The Select Control Block (SELCB), illustrated in the Figure below, is built whenever any SELECT or EXCLUDE commands are encountered. It is pointed to by the CARDOWN field in the CARCB. It contains such information as a table name containing DSNs to include or exclude, and a list of volumes to be included or excluded. In addition, it contains a pointer (SELDSN@) to the DSN Control Block (DSNCB) and a pointer (SELDOWN) to the CRITERIA Control Block (CRICB). The DSNCB contains a list of data set names entered by the user in the DSN parameter of the SELECT or EXCLUDE command. The CRICB contains the criteria comparison to be performed as specified in the CRITERIA parameter of the SELECT or EXCLUDE command.

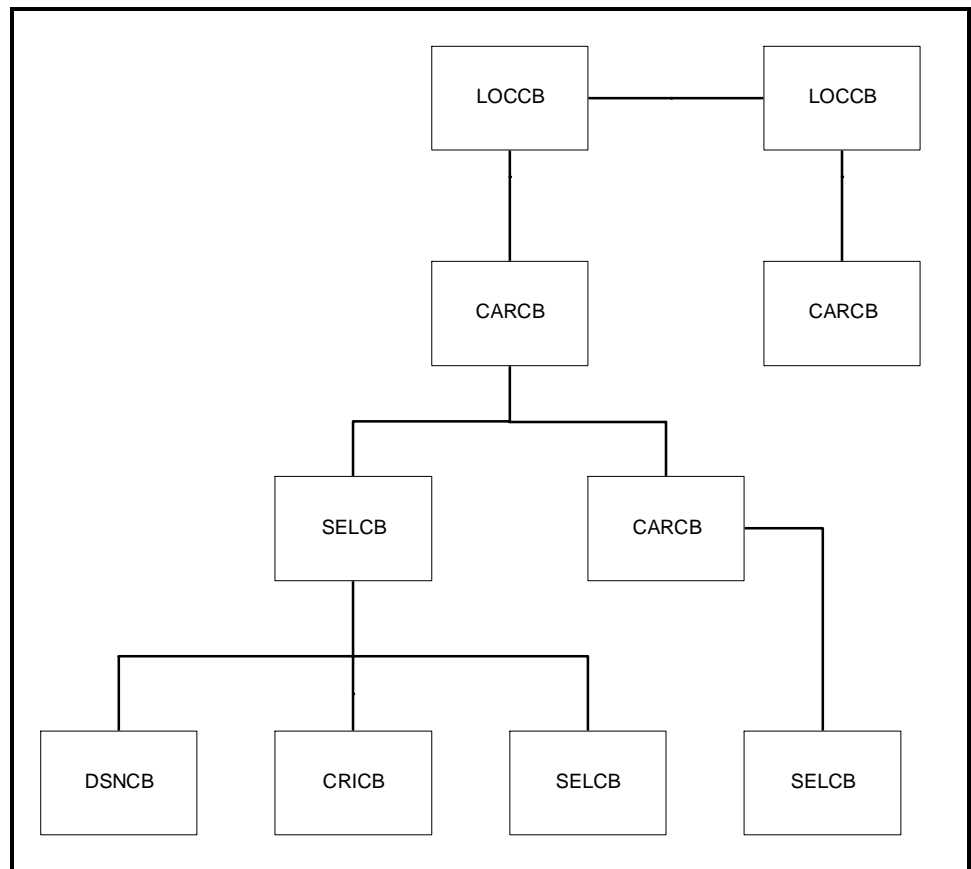


Figure 9-1. Control Block Sample Structure

The following Figure displays typical input commands that would build structure pictured above:

```

SCAN CATALOG=ADS050.AUSERCAT
SELECT DSN=(ADS.TJP/,ADS.DMS40.PMRFILE), CRITERIA=(TODAY-ARCDT,GE,5)
EXCLUDE TABLE=SAMPLE
BACKUP
SELECT VOLUME=ADS050
ARCHIVE RETPD=60 SCAN CATALOG=ADS050.SYSTMCAT
BACKUP
  
```

Figure 9-2. Sample Input Command Structure

Chapter 10. Security Processing

Many users are investing in computer hardware, software and security systems that meet the standards documented in the United States Department of Defense Trusted Computer System Evaluation Criteria (commonly called "*the Orange Book*"), reference number DOD 5200.28-STD.

This document provides a classification scheme for computer systems. The National Computer Security Center (NCSC) currently evaluates together as a Trusted Computer Base (TCB) combinations of computer hardware, and specific releases of operating system and reference monitor (security system).

Many users recognize that any code running APF-authorized can get into storage protection key 0 or the supervisor state, modify any part of memory, issue privileged instructions, bypass any or all parts of MSP system security, and basically do whatever it wants to. Thus the addition of APF-authorized utilities, such as many storage management systems, data base management systems and online transaction processors, may open up such a base system or TCB to exposures. However, at press time, the NCSC does not evaluate such utilities for a statement as to whether they degrade the integrity of the TCB, nor does it place such utilities on any evaluated product list.

Statement of Security Direction

It is our belief that when the security features described on the following pages are active, SAMS:Disk will not degrade the integrity of computer systems running on hardware, operating systems and reference monitors (security systems) that provide division C1, C2, or B1 protection.

Password protection is in effect only when not overridden by RACF or other security systems.

Security Overview

SAMS:Disk uses a multi-level approach to security. (A graphical representation of this approach follows this text.)

First, no special authorization is required to use SAMS:Disk — SAMS:Disk has the same security authorization as any other utility run by any of your users.

Second, SAMS:Disk uses separate address spaces for separate tasks. Thus, each task can be identified as a unique user (for example Logonid, user ACID or userid) by the security systems installed on your system (for example RACF, SECURE, or

ALERT). This provides one complete link between your SAMS:Disk users and system security.

Third, SAMS:Disk uses standard system interfaces. Before processing any data in a user data set, SAMS:Disk goes through the OPEN SVC of the operating system. Among other things, the OPEN SVC checks authorization with security systems installed on your system, such as RACF, SECURE, ALERT and OS- and VSAM- passwords.

To delete, define, catalog, uncatalog or recatalog user data sets, SAMS:Disk uses the SCRATCH and Catalog Management SVCs. These SVCs also check authorization with your security system(s). This protection is based upon your security systems, such as RACF, SECURE, ALERT and OS- and VSAM- passwords.

Fourth, SAMS:Disk provides an interface to RACF which is described on the following pages. This protection is based upon setting up system parameter run-time override protection, authorized Parmlib protection and your security system. This provides another complete link between your SAMS:Disk users and system security.

The integrity of your shop as a whole is, in turn, is based on the functions of your hardware and operating systems, and on the use of a security system to prevent unauthorized update of SYS1.PARMLIB, the authorized SAMS:Disk Parmlibs, SYS1.NUCLEUS, SYS1.SVCLIB, SYS1.LPALIB and all of your linklist and APF-authorized libraries.

If you have a security system in your shop, you will need to set up your production SAMS:Disk runs to have sufficient authority to process all data sets managed by SAMS:Disk. The simplest way to accomplish this is to run these SAMS:Disk tasks with a Logonid/user ACID/userid that has all authority to all data sets. Depending on your security system, other alternatives may also be possible. Turn to the topic *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide* for a discussion of some of these alternatives. Your individual users may process (archive, restore, migrate, and so on) those data sets for which they are authorized.

The following chart is a graphical representation of the multi-level approach to security used by SAMS:Disk.

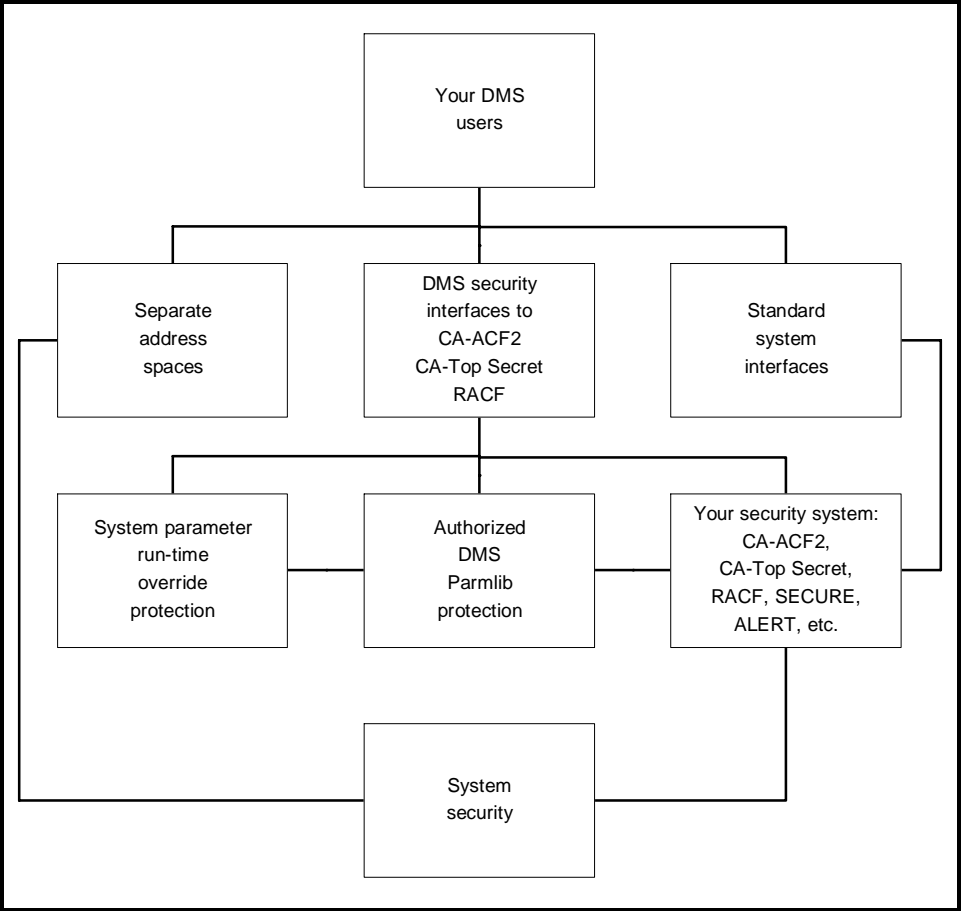


Figure 10-1. Multi-level approach to security.

Resources Protected by the SAMS:Disk Base System

The following resources are specifically protected by the base system of SAMS:Disk:

1. User data sets on DASD. This protection results from the use of standard system interfaces as described in the Security Overview discussion above.
2. SAMS:Disk archive data sets on DASD. You may optionally have your security package assign to all users the authority of NONE to all SAMS:Disk archive data sets on disk. This will prevent utilities other than SAMS:Disk from reading, modifying or deleting these archives. SAMS:Disk will take care of its own access to the archives. To determine the data set names of the SAMS:Disk archives, see the discussion under *"Naming the Archives"* on page 61 in this manual.

Unless you specify SAMS:Disk sysparm DISKPROT with a value of N, SAMS:Disk will also password-indicate each archive data set on disk, but without creating any passwords for these archive data sets on disk. This password-indication will also prevent utilities other than SAMS:Disk from reading, modifying or deleting these archives. SAMS:Disk will take care of its own access to the archives. This protection is selectable, and would supplement any protection from your security package.

3. SAMS:Disk archive data sets on tape. You may optionally have your security package assign to users who need to run backup, archive or merge to tape the authority of WRITE (UPDATE) and to all other users the authority of NONE to all SAMS:Disk archive data sets on tape. This will prevent utilities other than SAMS:Disk from reading, modifying or deleting these archives, except by authorized users. If you let SAMS:Disk sysparm TAPEPRTR default, SAMS:Disk Restore processing will take care of its own access to the archives. To determine the data set names of the SAMS:Disk archives, see the discussion under *"Naming the Archives"* on page 61 in this manual.

If you set sysparm TAPEPROT to a value of Y, SAMS:Disk will also password-indicate each archive data set on tape, but will not create any passwords for them. This will also prevent non-authorized users running utilities other than SAMS:Disk from reading, modifying or deleting the archives. SAMS:Disk will take care of its own access to the archives. This protection is selectable, and would supplement any protection from your security package.

4. Use of the //PARMLIB dd statement. Combined with the next item and the setting of SAMS:Disk sysparm SYSPARMO, this feature also protects the setting of SAMS:Disk sysparms. You may optionally

restrict SAMS:Disk such that SAMS:Disk will run only when it is using the one authorized parmlib, or one of a list of authorized parmlibs. For details on how to set up this restriction, see *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide*.

5. Use of SAMS:Disk sysparm overrides through use of the //SYSPARMS dd statement. Use of this dd statement is described in *"Overriding Sysparms Instream"* on page 104 in this manual. Users may override sysparms only if SYSPARMO is specified with a value of Y in the SYSPARMS member of your SAMS:Disk parmlib. For details on how to set up this restriction, see *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide*.
6. Use of the //MTFDEBUG dd statement. For details on how to set up this restriction, see *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide*.
7. SAMS:Disk PFD functions. Protection of SAMS:Disk PFD functions is documented under *"Defining PFD and DSCL User Options"* on page 67 of the *Installation Guide*.
8. Auto-restore of user data sets. Protection of the SAMS:Disk auto-restore function is documented under *"Installing the Auto-Restore Function"* on page 43 of the *Installation Guide*.

Resources Protected by the Security Interfaces

The following resources are specifically protected by the SAMS:Disk Security Interfaces:

- User data sets in the SAMS:Disk archives. Anyone with access to SAMS:Disk can access data sets in the SAMS:Disk archives unless you have installed a SAMS:Disk Security Interface. The SAMS:Disk Restore function goes through the OPEN SVC on the target data sets on disk, with OPEN checking authorization with your security package. But SAMS:Disk cannot go through the OPEN SVC on data sets in the archives, so authorization is not checked with your security package.

Similarly, in the Deferred Archive and Deferred Restore functions, SAMS:Disk cannot go through the OPEN SVC on data sets until the Dispose run, so authorization is not checked with your security package. This allows anyone to copy data sets in the archives by using the NEWNAME parameter on the RESTORE command or to archive or restore them in deferred mode, even if they would not otherwise be authorized to copy the data sets.

In addition, anyone with access to SAMS:Disk can delete (purge) the contents of the SAMS:Disk archives, unless you have installed a SAMS:Disk Security Interface. The SAMS:Disk files data set contains descriptions of data sets in the archives and of the archives themselves.

You can control the above exposures by installing a SAMS:Disk Security Interface and reviewing the setting of sysparms SECARVOL and SECFRVOL.

When an user issues a deferred restore, SAMS:Disk does not know whether or not the data set will be preallocated at the actual time of the restore. Whether or not the data set is preallocated will affect the level of authority that SAMS:Disk checks. The default level of SAMS:Disk authority that is checked at deferred restore time is OUTPUT. The level of authority checked at the time of the actual restore could be OUTPUT, SCRATCH or ALLOCATE. Sysparm SECURDRS allows the user to specify either OUTPUT, SCRATCH or ALLOCATE as the level of authority SAMS:Disk will check at the time of the deferred restore. For more detailed information, please turn to the sysparm description for *SECURDRS* on page 182 in this manual.

- Access to disk volumes. Anyone with access to SAMS:Disk can have access to disk volumes through the use of the volume-level functions of VBACKUP and VRECOVER, unless a SAMS:Disk Security Interface is installed and the sysparm SECURVOL defaults to a value of Y. Of special concern is the VRECOVER function, which can write over any volume, unless the SAMS:Disk security is properly set up.
- Reporting on user data sets. Anyone with access to SAMS:Disk can report on user data sets on disk, user data sets in the archives and on the archives themselves unless you have installed a SAMS:Disk Security Interface. The SAMS:Disk REPORT functions (except for the PDS directory reports) do not go through the OPEN SVC on user data sets; therefore, authorization is not checked with your security package. You can control this exposure by installing a SAMS:Disk Security Interface, specifying the SAMS:Disk sysparm SECURRPT with a value of Y.
- Auto-restore of user data sets. Protection of the SAMS:Disk auto-restore function is documented under *"Installing the Auto-Restore Function"* on page 43 of the *Installation Guide*.
- Command names. Anyone with access to SAMS:Disk can run any SAMS:Disk command unless you have installed a SAMS:Disk Security Interface. For example, your installation may choose to allow users to restore individual data sets with the RESTORE command, but allow only the data management group access to recover data sets with the implicit RECOVER command. You can control this exposure by installing a SAMS:Disk Security Interface, specifying the system

parameter SECURCMD with a value of Y, and creating appropriate entries in the *SECURDSN* member of the parmlib data set as described on page 559 in this manual.

- Subfiles of the FILES data set. Anyone with access to SAMS:Disk can access any subfile unless you have installed a SAMS:Disk Security Interface. For instance, your installation may choose to allow users to list information in the DSNINDEX subfile regarding data sets in the archives, but not allow them to list information in the ARCHVOLS subfile regarding the archives themselves. You can control this exposure by installing a SAMS:Disk Security Interface, specifying the system parameter SECURFIL with a value of Y, and creating appropriate entries in the *SECURDSN* member of the Parmlib data set as described on page 559 in this manual.

SAMS:Disk currently supports an interface to RACF. Although other security packages may be queried via the user exit SECURPEX.

Resources not Protected by SAMS:Disk

The following resources are not protected by SAMS:Disk processing:

- Auto-restore of user data sets in shops that do not have RACF. After an auto-restore completes, OPEN processing prevents user access to data sets to which they are not authorized. Therefore, there is no security hole, but in general anyone can cause the auto-restore of any user data set archived by SAMS:Disk except as described above.
- User data sets moved to tape by the Sequential Migrate function.
- SAMS:Disk program names. Some security packages allow you to set special authority rules for data sets when they are accessed through specified programs. In addition, you may have user exits on your operating system that may examine the program name field. This program name is taken from the JCL EXEC PGM parameter for batch functions and from the PFD select PGM parameter for PFD functions. The program name is stored in the CDE control block field CDNAME.

If you install SAMS:Disk as documented, all batch and PFD functions will run as the program name ADSMI002. SAMS:Disk TSS functions will each run with program names equal to their respective command names.

- The SAMS:Disk files data set itself. You will need to grant WRITE (UPDATE) authority to the files data set to users who run backup or archive functions, who queue archive or restore requests, or who do certain other functions.

You will need to grant WRITE (UPDATE) authority to the files data set to users who run the restore or recover functions with non-default values for sysparms RESIXRPD or RETEXCLD.

You may grant READ authority to the files data set to users who list the contents of the files data set, or who run the restore or recover functions with default values for sysparms RESIXRPD and RETEXCLD.

- The SAMS:Disk parmlib data set itself. You will need to grant READ authority to the parmlib data set to users who run any SAMS:Disk function. You will need to grant WRITE (UPDATE) authority to the parmlib data set to users who need to update sysparm settings, exemption tables, etc.
- The SAMS:Disk LOAD data set. Since the load library is APF-authorized, you may wish to give your users execute or READ authority only to this data set.
- Other SAMS:Disk distribution libraries.
- SAMS:Disk work data sets (RACF users see sysparm description for *RACFPDSW* on page 168 in this manual).

You should consider these resources as part of your security plan.

Also, note that SAMS:Disk, even without a security interface, will honor the ADSP attribute for RACF during restore or any other data set allocation. In this case, if ADSP is turned on, the DS1RACF bit will be set on. SAMS:Disk will honor the NOADSP attribute as well. If the userid submitting the job has the NOADSP attribute, SAMS:Disk will maintain the original setting of the DS1RACF bit.

Security and the SAMS:Disk Auto-Restore Function

As discussed in "*Installing the Auto-Restore Function*" beginning on page 43 of the *Installation Guide*, since the SAMS:Disk auto-restore started task cannot be identified by a JOB statement or a LOGON procedure, most security packages have a Started Procedures Table to associate a started task name with a Logonid/user ACID/userid and possible group name.

If your security package protects resources accessed by started tasks, and you have installed or will install the SAMS:Disk auto-restore function, and you would like to auto-restore data sets protected by your security package, you must create an entry in your Started Procedures Table to associate the SAMS:Disk auto-restore started task name DMSAR with a Logonid/user ACID/userid and possible group name.

There is one exception to that last statement — you will probably not have to create an entry if you install the SAMS:Disk RACF Security Interface, specify sysparm ARSECUREy, and specify neither sysparm SECURFIL nor SECURCMD.

SAMS:Disk Security Interface General Information

The SAMS:Disk Security Interfaces are designed to meet these goals:

- To provide the same level of security to resources unique to SAMS:Disk that your security package would normally provide to equivalent resources which, if left unprotected, might be regarded as security exposures.
- To pre-check authorization. This means that SAMS:Disk checks authorization before executing a system function, such as OPEN, that will also check authorization. Pre-checking results in two advantages:
 - a. The messages are often clearer. SAMS:Disk can issue an authorization-type message rather than a function-type message with return codes and reason codes.
 - b. Processing is often saved. For instance, if the archive function knows it does not have SCRATCH authority to a user data set, it can stop the archive before opening the data set, mounting a tape, backing up the data, etc.
- To process discrete RACF profiles. Because discrete RACF profiles are deleted when data sets referenced are scratched, special processing is often needed.

For more details on this goal, turn to the topic *"RACF Discrete Profile Handling"* on page 393 in this manual.

- To be auditable.

The exposure-closing and pre-checking goals are accomplished by querying your systems security package for approval prior to accessing a resource. With RACF, SAMS:Disk issues the RACROUTE REQUEST=AUTH macro.

SAMS:Disk then examines the return code from your security package. If authorized, SAMS:Disk continues with the processing of that resource. If not authorized, SAMS:Disk prints an appropriate message and bypasses that resource. Either way, processing continues to the next resource as required by the SAMS:Disk commands.

In addition to the SAMS:Disk authorization failure message, by default each security package issues its own authorization failure message to the system log, the security console, and to the TSS user or job log. It may also create SMF records for a report writer indicating access violations. If you wish to inhibit RACF from issu-

ing their own authorization failure messages or creating SMF records during queries from their respective SAMS:Disk Security Interfaces running as APF-authorized tasks, specify the sysparm SECURLOG with a value of N.

Regardless of the setting of sysparm SECURLOG, each security package will continue to issue its own authorization failure messages for failures occurring outside the SAMS:Disk Security Interfaces while running SAMS:Disk functions.

The auditability goal is implemented by SAMS:Disk internal diagnostic system parameter DIAGAUTH. If this sysparm is specified with a value of Y, SAMS:Disk will issue additional messages regarding exactly what the SAMS:Disk Security Interface is passing to your security system, and what your security package returns. For RACF, SAMS:Disk also prints the profile status of data sets processed, and messages regarding processing of discrete profiles by SAMS:Disk. This sysparm is very useful in diagnostic situations.

Running Without a SAMS:Disk Security Interface

If you run SAMS:Disk without a Security Interface installed or run with your Security Interface turned off, SAMS:Disk will not do any authority checking and will not process discrete RACF profiles. You will get the benefits of the authorization checking done by the operating system described in the *"Security Overview"* section on page 369 but you will have the exposures listed in the sections entitled *"Resources Protected by the Security Interfaces"* on page 373 and *"Resources Not Protected by SAMS:Disk"* on page 375 in this manual.

Password-Indicated Data Set Support

This topic is presented in *"Step 1. Select Password-Indicated Data Set Support Options"* on page 18 of the *Installation Guide*.

What Authority Does SAMS:Disk Look For?

For any security protection to be effective, you must have activated the interface as described in *"Activating SAMS:Disk Security Features and Interfaces"* on page 18 of the *Installation Guide*. If you do not have a SAMS:Disk Security Interface installed, or a resource is not protected in your environment, SAMS:Disk assumes that it has sufficient authority and will process that resource.

SAMS:Disk performs authorization checking on volumes and data sets to ensure that you have sufficient authority to those resources. Sufficient authority depends on the function that you are using.

In order to assist you in determining the authority for which SAMS:Disk will query your security package, we have provided a series of tables. You should use only the set of tables that corresponds to your security package.

Each set of tables contains three types of tables, as follows:

1. Protection of SAMS:Disk Subfiles

This table describes the security authority level needed to access subfiles in the files data set for each SAMS:Disk function.

2. Protection of SAMS:Disk Commands

This table describes the security authority level needed to use the commands for each SAMS:Disk function.

3. Protection of User Data Sets

This table describes the security authority level needed to access user data sets for each SAMS:Disk function.

SAMS:Disk can check authority both to the data set and its volume. In order to have the RACF check for volume level authority, sysparm SECURVOL must default to a value of Y.

For data set-level functions, if SECURVOL defaults to a value of Y, SAMS:Disk checks volume authority first. If volume authority is approved, the function is performed with no further checking. Otherwise SAMS:Disk continues with data set authority checking.

For volume-level functions, such as VBACKUP and VRECOVER, SAMS:Disk only checks for volume-level authority. If volume authority is approved, the function is performed with no further checking. Otherwise, the function is disallowed.

Authority Level Translation Tables for RACF

In this section are three tables to assist you in determining the level of authority for which SAMS:Disk will query your RACF security package.

The first table, Protection of SAMS:Disk Subfiles under RACF, describes the RACF authority levels and the SAMS:Disk internal authority levels needed for access to subfiles in the files data set.

Please note that security validation will be done on SAMS:Disk subfiles only if sysparm SECURFIL has been specified with a value of Y and an appropriate entry exists in the parmlib member SECURDSN. If this is not the case, no security validation is done and access is authorized to the subfile.

Down the left-hand side of the table, you will find the SAMS:Disk functions, and across the top of the table are the names of subfiles within the files data set. The table indicates the authority level needed to access each subfile for each SAMS:Disk function. The RACF authority level is shown first. The corresponding SAMS:Disk internal authority level is enclosed in parentheses directly beneath the RACF authority level.

Following is an example of how to use this table to determine the security level authorization needed by the ARCHIVE function to access the subfiles used during ARCHIVE processing:

1. Locate the ARCHIVE function in the left-hand column.
2. Read across the column headings to identify the subfiles used by ARCHIVE.

You will see that SAMS:Disk will check for the RACF authority level of UPDATE to access both the DSNINDEX and the ARCHVOLS subfiles. In addition, SAMS:Disk will check for the RACF authority level of READ for the ARCHVOLS subfile. The RACF authority level of UPDATE translates to the SAMS:Disk internal authority level of OUTPUT. The RACF authority level of READ translates to the SAMS:Disk internal authority level of INPUT.

Function	SUBFILES					
	DSNINDEX	ARCHVOLS	MIGRECAT	DASDSPCB	ARCHCMDS	RESTCMDS
Archive	Update (output)	Update (output) Read (input)				
Backup	Update (output)	Update (output) Read (input)				
Restore	Write (update) Read (input)	Read (input)				
Auto-- Restore	Write (update) Read (input)	Read (input)				
Recover	Write (update) Read (input)	Read (input)				
Vbackup	Update (output)	Update (output) Read (input)				

Function	SUBFILES					
	DSNINDEX	ARCHVOLS	MIGRECAT	DASDSPCB	ARCHCMDS	RESCMDS
Vrecover	Read (input)	Read (input)				
Listd	Read (input)	Read (input)				
Ixmaint	Update (output)	Update (output) Read (input)				
Ixupdate	Update (output)	Update (output) Read (input)				
Merge	Update (output) Read (input)	Update (output) Read (input)				
Rebuild	Update (output)	Update (output)				
Seq. Migrate			Update (output)			
Billing				Update (output)		
Listreq					Read (input)	Read (input)
Derase					Update (output)	Update (output)
Format	ALTER (scratch) on each subfile to be formatted over					
Unload	Read (input) on each subfile					
Reload	ALTER (scratch) on each subfile to be formatted over and Update (output) on each subfile in the unloaded DSNAME					
Reorg	Read (input) on each subfile; ALTER (scratch) on each subfile; Update (output) on each subfile in the unloaded DSNAME					
Racfchk1	Read (input) on each subfile					

Relevant system parameters:
RESIXRPD

Protection of Commands Under RACF

The following table, Protection of SAMS:Disk Commands under RACF, describes the RACF and SAMS:Disk internal authority levels needed for access to SAMS:Disk commands.

Please note that security validation will be done on SAMS:Disk commands only if sysparm SECURCMD is specified with a value of Y and an appropriate entry exists in the parmlib member SECURDSN. If this is not the case, access is authorized for the SAMS:Disk command.

The table is organized into three columns. The first column lists the SAMS:Disk functions, the second lists the commands used with that SAMS:Disk function and the third column specifies the authority level needed for the SAMS:Disk command. The RACF authority level is shown first, followed by the corresponding SAMS:Disk internal authority level, which is enclosed in parentheses.

Following is an example of how to use this table to determine the security level authorization needed to use the VOLDELETE command for the Archive Management function:

1. Locate Archive Management in the Function column.
2. Read across to the Command column and find the VOLDELETE command.
3. Then read across to the Authority Level column to see that the RACF authority level UPDATE is needed to use the command.

This translates to the SAMS:Disk internal authority level of OUTPUT.

Function	Command	Authority Level
DSCL	ARCHIVE BACKUP BILLING COPY DELETE EXCLUDE FIND MOVE PASSWORD RELEASE REPORT SCAN SELECT SET VBACKUP	Update (output)
SMF Report	SELECT	Update (output)
Explicit Archive	ARCHIVE DARCHIVE MODESET PASSWORD	Update (output)

Function	Command	Authority Level
Restore/Recover	DRESTORE PASSWORD RESTORE RECOVER VRECOVER	Update (output)
Deferred Archive/Restore	DERASE LISTREQ	Update (output) Read (input)
PDS Compress	COMPRESS SCAN RELOAD	Update (output)
Sequential Migrate to Tape	MIGRATE MODESET RECATALOG SCAN	Update (output)
DASD Space Billing	ACCUMULATE BILLING EXTEND RENAME SELECT	Update (output)
Archive Management	ANALYZE BUILDARC CHANGE COPY DELETE DSNDELETE EXTRACT LISTD LISTV MERGE PURGE REBUILD RELOAD RESET TAPEFIX TRECOVER UNLOAD VALIDATE VOLDELETE	Update (output)
Files Management	FILEDUMP FRECOVER LOADLIB PROCESS REPARCH RETAPE	Update (output)

Function	Command	Authority Level
Other diagnostics	DUMP RMAP	Update (output)
Install product	CATALOG DSCB SET	Update (output)
Tapepool Utilities	ADD DELETE ERASE LIST	Update (output)
Threshold Manager	EXCLUDE= GLOBAL= INCLUDE= MULTIVOL= POOLDEF=	Not checked
RACF Utilities	CHECK CREATE DELETE DISPLAY RENAME SELECT	Update (output)
Non documented Utilities	BLDL COPY DISPLAY DISPOSE DMSGLOBAL INIT MIGRATE PASSWORD REPORT RETAIN SCAN SELECT SEARCH VREPORT	Update (output)

Protection of User Data Sets Under RACF

The following tables, Protection of SAMS:Disk Data Sets Under RACF, describe both the RACF authority levels and the SAMS:Disk internal authority levels needed for access to data sets and their volumes. Each SAMS:Disk function appears in a separate table.

Each function table is organized into three major columns. The first column lists the SAMS:Disk functions. The second column shows the breakdown between volume and data set level security checks. The third column shows the authority level needed to access the volume or data set. Depending on the SAMS:Disk function,

the third column is further divided into command parameters that can affect the authorization level needed to access the resource. The RACF authority level is shown first, followed by the corresponding SAMS:Disk internal authority level, which is enclosed in parentheses. The words no check are used if there is to be no security checking done.

Please note that SAMS:Disk will check access to a volume only when sysparm SECURVOL is specified with a value of Y. When volume-level authorization checking is done (SCRATCH action only), it will always precede the data set-level check. If access to the volume is authorized, the data set-level authorization checking will be bypassed. Data set-level authorization checking is therefore done only if any of these are true:

- a. sysparm SECURVOL is specified with a value of N
- b. the volume is not protected by your security interface
- c. access to the volume is denied

Following is an example of how to use the table to determine the security level authorization needed for RESTORE processing:

The sample RESTORE command is:

```
RESTORE DSNAME=SBKSV.DATA,NEWNAME=SBKSV.DATA.NEW,SCRATCH
```

The target data set, SBKSV.DATA.NEW, is preallocated. Sysparm SECURVOL is set to Y.

1. Find the page of the table describing the RESTORE function.

To find the RACF security level authorization needed to access SBKSV.DATA, first locate the data set to be restored section in the Resource column.

2. Read across the volume-level check line to see that no volume level checking is done. Since no volume check is done, read across the data set-level line to the authority level columns. They show that if the SCRATCH and NEWNAME parameters are specified on the command, RACF authority levels of UPDATE and READ are needed to access SBKSV.DATA.

To find the RACF security level authorization needed to access SBKSV.DATA.NEW, first locate the target data set preallocated section in the Resource column. Read across the volume level check line to the authority level columns. They show that the RACF authority level of ALTER is needed if the SCRATCH parameter is specified on the command. If the volume security check fails or is not done, and if the SCRATCH and NEWNAME parameters are specified on the

command, RACF authority levels of ALTER and UPDATE are needed to access SBKSV.DATA.NEW.

Protection of User Data Sets Under RACF — Reports

Resource to be checked	Volume or data set level check	Authority level
Data set to be reported on	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Read (report) Control (control)

Relevant RACF system parameters:

SECURRPT
VSACCESS

Protection of User Data Sets Under RACF — Archive

Resource to be checked	Volume or data set level check	Authority level		
		With each command	Catalog action specified	Newname parm specified
Data set to be archived	Volume level check (if SECURVOL "Y")	Alter (scratch)	Alter (scratch)	no check
	Data set level check (done if vol check fails or not done)	Read (input) Alter (scratch)	Read (input) Alter (scratch)	no check
Cluster to be archived	Volume level check (if SECURVOL "Y")	Alter (scratch)	no check	no check
	Data set level check (done if vol check fails or not done)	Read (input) or Control (control) plus Alter (scratch & catalog)	Read (input) or Control (control) plus Alter (scratch & catalog)	no check

Resource to be checked	Volume or data set level check	Authority level		
		With each command	Catalog action specified	Newname parm specified
New data set name	Volume level check if SECURVOL "Y"	no check	no check	no check
	Data set level check (done if vol check fails or not done)	no check	no check	Update (rename to)

Relevant RACF system parameters:

RACFBKUP
RACFNEWN
VSACCESS

Protection of User Data Sets Under RACF — Backup

Resource to be checked	Volume or data set level check	Authority level
Data set to be backed up	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Read (input) or Control (control)

Relevant RACF system parameters:

RACFBKUP
VSACCESS

Protection of User Data Sets Under RACF — Restore

Resource to be checked	Volume or data set level check	Authority level	
		Scratch parm specified	Scratch parm not specified
Data set to be restored	Volume level check (if SECURVOL "Y")	no check	no check
	Data set level check (done if vol check fails or not done)	Update (output) ... or Control (control)	Update (output) ... or Control (control)

Target data set preallocated	Volume level check (if SECURVOL "Y")	Alter (scratch)	no check
	Data set level check (done if vol check fails or not done)	Alter (scratch)	Update (output)
Target data set not preallocated (checks preexisting, non-SAMS:Disk saved, profiles only)	Volume level check if SECURVOL "Y"	no check	no check
	Data set level check (done if vol check fails or not done)	based on value of sysparm RACFALLO which defaults to Alter (alloc)	based on value of sysparm RACFALLO which defaults to Alter (alloc)

Relevant RACF system parameters:

RACFALLO
 RACFMODL
 RACFNEWN
 RACFPRED
 SECURDRS
 VSACCESS

Protection of User Data Sets Under RACF — LISTD

Resource to be checked	Volume or data set level check	Authority level
Data set to be reported on	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Read (report)

Protection of User Data Sets Under RACF — IXMAINT

Resource to be checked	Volume or data set level check	Authority level
Data set to be scratched from the SAMS:Disk archives	Volume level check (if SECURVOL "Y")	Alter (scratch)
	Data set level check (done if vol check fails or not done)	Alter (scratch)

Protection of User Data Sets Under RACF — MERGE

Resource to be checked	Volume or data set level check	Authority level
Data set to be scratched from the SAMS:Disk archives	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Update (output)

Protection of User Data Sets Under RACF — REBUILD

Resource to be checked	Volume or data set level check	Authority level
Data set to be overlaid in the SAMS:Disk archives	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Update (output)

Protection of User Data Sets Under RACF — MOVE/COPY

Resource to be checked	Volume or Data set level check	Authority level
Data set to be copied	volume level check (of SECURVOL "Y")	no check
	data set level check (done if vol check fails or not done)	Update (output) or Control (control)
Data set to be moved	volume level check (of SECURVOL "Y")	Alter (scratch)
	data set level check (done if vol check fails or not done)	Alter (scratch)
Target data set preallocated	volume level check (of SECURVOL "Y")	no check
	data set level check (done if vol check fails or not done)	Update (output)

Resource to be checked	Volume or Data set level check	Authority level
Target data set not preallocated (checks preexisting, non-SAMS:Disk, saved profiles only)	volume level check (of SECURVOL "Y")	no check
	data set level check (done if vol check fails or not done)	based on value of sysparm RACFALLO which defaults to Alter (alloc)

Relevant RACF system parameters:

RACFALLO
RACFNEWN
RACFVCAV
VSACCESS

Protection of User Data Sets Under RACF — RELEASE

Resource to be checked	Volume or Data set level check	Authority level
Data set to have space released	volume level check (of SECURVOL "Y")	no check
	data set level check (done if vol check fails or not done)	Update (output)

Protection of User Data Sets Under RACF — Compress

Resource to be checked	Volume or data set level check	Authority level		
		USE parm specified	USE parm not specified	RELOAD command specified
Data set to be compressed	Volume level check (if SECURVOL "Y")	no check	Alter (scratch)	no check
	Data set level check (done if vol check fails or not done)	Read (input) Alter (scratch)	Read (input) Update (output) Alter (scratch)	no check

Resource to be checked	Volume or data set level check	Authority level		
		USE parm specified	USE parm not specified	RELOAD command specified
Data set to be reloaded	Volume level check (if SECURVOL "Y")	no check	no check	no check
	Data set level check (done if vol check fails or not done)	no check	no check	Update (output)

Relevant RACF system parameters:
RACFPDSW

Protection of User Data Sets Under RACF — Sequential Migrate

Resource to be checked	Volume or data set level check	Authority level	
		Noscratch parm specified	Noscratch parm not specified
Data set to be migrated	Volume level check (if SECURVOL "Y")	no check	Alter (scratch)
	Data set level check (done if vol check fails or not done)	Read (input)	Read (input) Alter (scratch)

Relevant RACF system parameters:
RACFSEQM

Relevant RACF SAMS:Disk user exits:
RACFSMEX

Protection of User Data Sets Under RACF — Delete

Resource to be checked	Volume or data set level check	Authority level
Data set to be scratched	Volume level check (if SECURVOL "Y")	Alter (scratch)
	Data set level check (done if vol check fails or not done)	Alter (scratch)

Protection of User Data Sets Under RACF — LISTREQ

Resource to be checked	Volume or data set level check	Authority level
Data set to be reported on	Volume level check (if SECURVOL "Y")	no check
	Data set level check (done if vol check fails or not done)	Read (report)

Protection of User Data Sets Under RACF — DERASE

Resource to be checked	Volume or data set level check	Authority level
Data set to have request deleted	Volume level check (if SECURVOL "Y")	Alter (scratch)
	Data set level check (done if vol check fails or not done)	Alter (scratch)

Protection of User Data Sets Under RACF — DSCB Update

Resource to be checked	Volume or data set level check	Authority level
Data set to be processed	Volume level check (if SECURVOL "Y")	Alter (scratch)
	Data set level check (done if vol check fails or not done)	Update (output) Alter (scratch)

Protection of User Data Sets Under RACF — VBACKUP

Resource to be checked	Authority level
Volume to be backed up	Read (input)

Protection of User Data Sets Under RACF — VRECOVER

Resource to be checked	Authority level
Volume to be recovered	Read (input)
Target volume	Alter (scratch) Update (output)
IDVOL for CLIP	Alter (scratch) Update (output)

Functions that Need no Data Set/Volume Authorization

The following functions do not actually access any user data sets, so no security authorization checking is done at the data set/volume level. However they do use commands and do access the files data set, so security authorization checking is done only at the files data set subfile level and the SAMS:Disk command level.

- Format files data set
- Unload files data set
- Reload files data set
- Reorg files data set

RACF Discrete Profile Handling

RACF Terms Defined

DISCRETE PROFILE

A discrete profile is a security mask covering one resource such as a data set on a particular volume. A discrete profile is deleted by the operating system when the resource covered by that profile is deleted. The profile resides as a record in the RACF data set.

GENERIC PROFILE

A generic profile is a security mask covering zero or more resources, such as all data sets on all volumes that have a data set name matching a particular pattern. A generic profile is not deleted by the operating system when a resource covered by that profile is deleted. The profile resides as a record in the RACF data set.

RACF-INDICATED

A data set is RACF-indicated if the RACF indicator bit for the data set is turned on. For non-VSAM data sets, the RACF indicator bit is the DS1IND40 bit (bit x'40' at offset 93 x'5D') located in the format-1 DSCB. For VSAM data sets, the RACF indicator bit is the x'80' bit in the SECF field located in the cluster's catalog entry.

ALWAYS-CALL

A feature of the operating system where a security package is queried for authorization regardless of whether the resource is RACF-indicated or not.

RACF-PROTECTED

Always call is on: A RACF-indicated data set covered by a discrete or generic profile is protected. A RACF-indicated data set not covered by a discrete or generic profile is in error. A non-RACF-indicated data set covered by a generic profile is

protected. A non-RACF-indicated data set not covered by a generic profile is not protected. Always call is off: A RACF-indicated data set covered by a discrete or generic profile is protected. A RACF-indicated data set not covered by a discrete or generic profile is in error. A non-RACF-indicated data set is not protected.

SAMS:Disk-*SAVED* PROFILE

When SAMS:Disk backs up or archives a data set, SAMS:Disk will optionally copy the discrete profile, if any, protecting that data set. This copied or SAMS:Disk-saved profile contains all of the original profile information, except that the data set name and volume have been changed (SAMS:Disk correlates this saved profile back to its DSNINDEX entry via the RACFENCD subfile, described beginning on page 395 in this manual).

The copied profile resides in the RACF data set, just as any other RACF profile.

The SAMS:Disk-saved profile is then available to put back the original profile information should the data set be restored or recovered at a later time.

Generic profiles are not scratched by RACF when resources covered by them are scratched. Therefore, SAMS:Disk does not copy generic profiles, only discrete profiles.

RACF and VSAM Data Sets

Like non-VSAM data sets, VSAM clusters may also be RACF-protected. SAMS:Disk maintains RACF profiles for VSAM clusters in the same way support is provided for non-VSAM data sets.

Consistent with Fujitsu's policy, SAMS:Disk maintains discrete profiles for the cluster names of VSAM clusters in EDF-Catalogs. Consistent with Fujitsu's policy, SAMS:Disk maintains discrete profiles for each of the cluster, data component, index component, and any AIX or PATH names of VSAM clusters in non-EDF-Catalogs. SAMS:Disk maintains this policy even when a cluster is transferred back and forth between the two different types of catalogs.

Backing Up RACF Discrete Profiles (RACFBKUP)

In normal processing, any SAMS:Disk function that archives or backs up a data set protected by a discrete profile will also create a SAMS:Disk-saved profile for the data set. For some users this may be costly in terms of both job performance and DASD space utilization in the RACF profile data set.

If you do not want SAMS:Disk to create SAMS:Disk-saved profiles for back ups of data sets protected by discrete profiles, specify sysparm RACFBKUP with a value of Y. SAMS:Disk will create SAMS:Disk-saved profiles for archives of data sets protected by discrete profiles.

If you do not want SAMS:Disk to create SAMS:Disk-saved profiles for back ups nor archives of data sets protected by discrete profiles, specify the RACFBKUP system parameter with a value of F.

If you specify a value of Y or F, it is your responsibility to create a correct profile for data sets being restored prior to using the data set. Review the RACFMODL discussion below as one method of creating discrete profiles to cover data sets being restored for which SAMS:Disk has no associated SAMS:Disk-saved profiles.

Also please review the sysparm RACFPRED regarding its effect on restoring data sets where a discrete profile already exists.

Processing RACF-Indicated Data Sets Without Profiles

SAMS:Disk will process RACF-indicated data sets for which no profile exists when the SAMS:Disk RACF Security Interface is inactive (sysparm RACFSUPPn is specified). SAMS:Disk will issue an error message and bypass RACF-indicated data sets for which no profile exist when the SAMS:Disk RACF Security Interface is active (sysparm RACFSUPPy is specified).

Restoring RACF-Indicated Data Sets Having no Profiles

To protect data set access integrity, SAMS:Disk normally will not allow RACF-protected data sets to be restored unless either their associated SAMS:Disk-saved profiles are available or the data sets were protected by a generic profiles when backed up.

If you specify sysparm RACFSUPPn, RACFBKUPy or RACFBKUPf at archive/backup time, suffer loss of some SAMS:Disk-saved profiles, or go to a disaster recovery site without your RACF data set, you need some way to restore your data sets that were originally protected by discrete profiles.

This option allows you to specify a default discrete profile to be used for all data sets to be restored that do not have profiles.

The default profile is specified through sysparms that contain a data set name and volume serial number. The values must correspond to an existing discrete RACF profile.

When a data set without a SAMS:Disk-saved profile is to be restored, the attributes and access list of the model profile are assigned to the restored data set. Normally, the attributes are those of the RACF administrators or some other individual responsible for maintenance of RACF profiles. They may also be used as codes to be passed to RACF user exits that assign attributes based on installation requirements.

To activate the use of model profiles, build a discrete profile to be used as a model, specify sysparm RACFMODL with a value of Y, sysparm RACFMDSN with the 1- to 44-character name of the model profile, and sysparm RACFMVOL with the 1- to 6-character volume association of the model profile.

Reconciling SAMS:Disk and RACF Records

The RACF encode type 2 technique requires SAMS:Disk to keep in synchronization three different records relating to saved RACF profiles. First, the DSNINDEX subfile record in the files data set must contain the bit to indicate that a profile is being maintained for the archived data set. Then RACF profile records must be kept in the users RACF profile data set. Third, a RACFENCD subfile record must be in the files data set relating the DSNINDEX record to the RACF profile.

If SAMS:Disk is improperly set up and run, the relationship between some records may be destroyed. For example, running the IXMAINT utility to delete expired data sets without the sysparms RACFPROC and RACFSUPP specified with a value of Y will leave disconnected RACF profiles on the RACF profile data set and will leave unused RACFENCD records in the files data set. These disconnected records will not cause any harm except to take up space in the data sets.

To verify that this situation does not exist, or to correct it if it does, the following utility reconciles the relationship between RACF profile information. The utility verifies the names of SAMS:Disk RACF profiles read from the users RACF data set with the RACFENCD record and the RACF bit in the DSNINDEX record. The following conditions are processed:

- If a valid profile exists, but the corresponding RACFENCD record does not exist, delete the RACF profile.
- If a RACFENCD record exist for a DSNINDEX record, but a RACF profile does not exist, delete the RACFENCD record and turn off the DSNINDEX record bit.
- If both a valid profile and a RACFENCD record exist for a DSNINDEX record, but the DSNINDEX record bit is off, turn on the DSNINDEX record bit.
- If both a valid profile and a RACFENCD record exist for a DSNINDEX record, but the DSNINDEX record does not exist, delete both the RACF profile and the RACFENCD record.
- If a DSNINDEX record bit is on, but the corresponding RACFENCD record does not exist, turn off the DSNINDEX record bit. Use this utility whenever the integrity of the relationship between profile information is in question. Because of the need to process all SAMS:Disk-saved profiles, the user ID of the job statement of this utility must have the SPECIAL attribute.

This utility can take a significant amount of time to run, particularly in installations with a large RACF data set.

The program runs as a batch application. Use caution when executing this utility to ensure no valid records are deleted by mistake. The first step of the utility contains an unload of the files data set to ensure a good backup copy will always be avail-

able. This utility should be run in simulate mode and all return codes should be checked thoroughly to ensure the integrity of the DSNINDEX, RACFENCD subfiles and Fujitsu's RACF profile data set.

The utility uses the following RACF TSS command to create the list of SAMS:Disk-saved profiles for reconciling with the RACFENCD and DSNINDEX records:

```
SEARCH MASK(racfusid.,.THM) NOGENERIC
```

The lower case letters represent the values of the SAMS:Disk sysparm. Information on the command itself can be obtained by entering the TSS command HELP SEARCH. Thus the possibility exists that some profiles that are not SAMS:Disk-saved profiles may be listed and processed:

- If you have any discrete RACF profiles that are not SAMS:Disk-saved profiles that have a data set name that starts with the value of sysparm RACFUSID, followed by a period, followed somewhere later by .THM, then those profiles will be listed by this utility, found to have no corresponding entry in the RACFENCD subfile, and if the XPRO parameter is specified, deleted. It is very unlikely that any non-SAMS:Disk-saved profiles would meet these criteria and would be deleted by this utility, but you should consider this possibility.

Note for Users With Multiple Files Data Sets: If some or all of your files data sets have the same values for the sysparms RACFUSID and RACFDVOL, you must not use the XPRO parameter of this utility. SAMS:Disk uses the values of these sysparms to determine which profiles are SAMS:Disk-saved profiles. Running the XPRO parameter on this utility on a files data set will delete profiles saved for other files data sets if and only if the other files data sets have the same value of the sysparms RACFUSID and RACFDVOL. If you have only one files data set or all of your files data sets have Parmlibs with different values for RACFUSID or RACFDVOL, you are free to use any or all parameters of this utility.

Following is sample JCL for executing the utility.

```
// EXEC RACFCHK1
//RECONCI.SYSIN DD *
CHECK SIM,OKL,XPRO,XENC,XDSN
```

Check Command Parameter Description

```
CHECK SIMULATE,XPROFILES,XENCDCRECS,XDSNINDEX,OKLIST
```

SIMULATE

Specification of this parameter causes a report to be produced indicating the data selected and the actions that would have occurred in live mode processing. It is useful in verifying that the input parameters will actually produce the intended results. It is highly recommended that this parameter be used before running in live mode.

XPROFILES

If a valid profile exists, but the corresponding RACFENCD record does not exist, delete the RACF profile.

If you do not specify this parameter, this condition will not be listed or corrected. Before using this parameter, read carefully the warning above.

XENCDCRECS

If a RACFENCD record exist for a DSNINDEX record, but a RACF profile does not exist, delete the RACFENCD record and turn off the DSNINDEX record bit.

If you do not specify this parameter, this condition will not be listed or corrected.

XDSNINDEX

If both a valid profile and a RACFENCD record exist for a DSNINDEX record, but the DSNINDEX record bit is off, turn on the DSNINDEX record bit.

If both a valid profile and a RACFENCD record exist for a DSNINDEX record, but the DSNINDEX record does not exist, delete both the RACF profile and the RACFENCD record.

If a DSNINDEX record bit is on, but the corresponding RACFENCD record does not exist, turn off the DSNINDEX record bit.

If you do not specify this parameter, these conditions will not be listed or corrected.

OKLIST

Specify this parameter to print messages for each valid relationship processed. If you do not specify this parameter, messages will be printed only for error conditions.

RACF Checking Logic Summary in SAMS:Disk

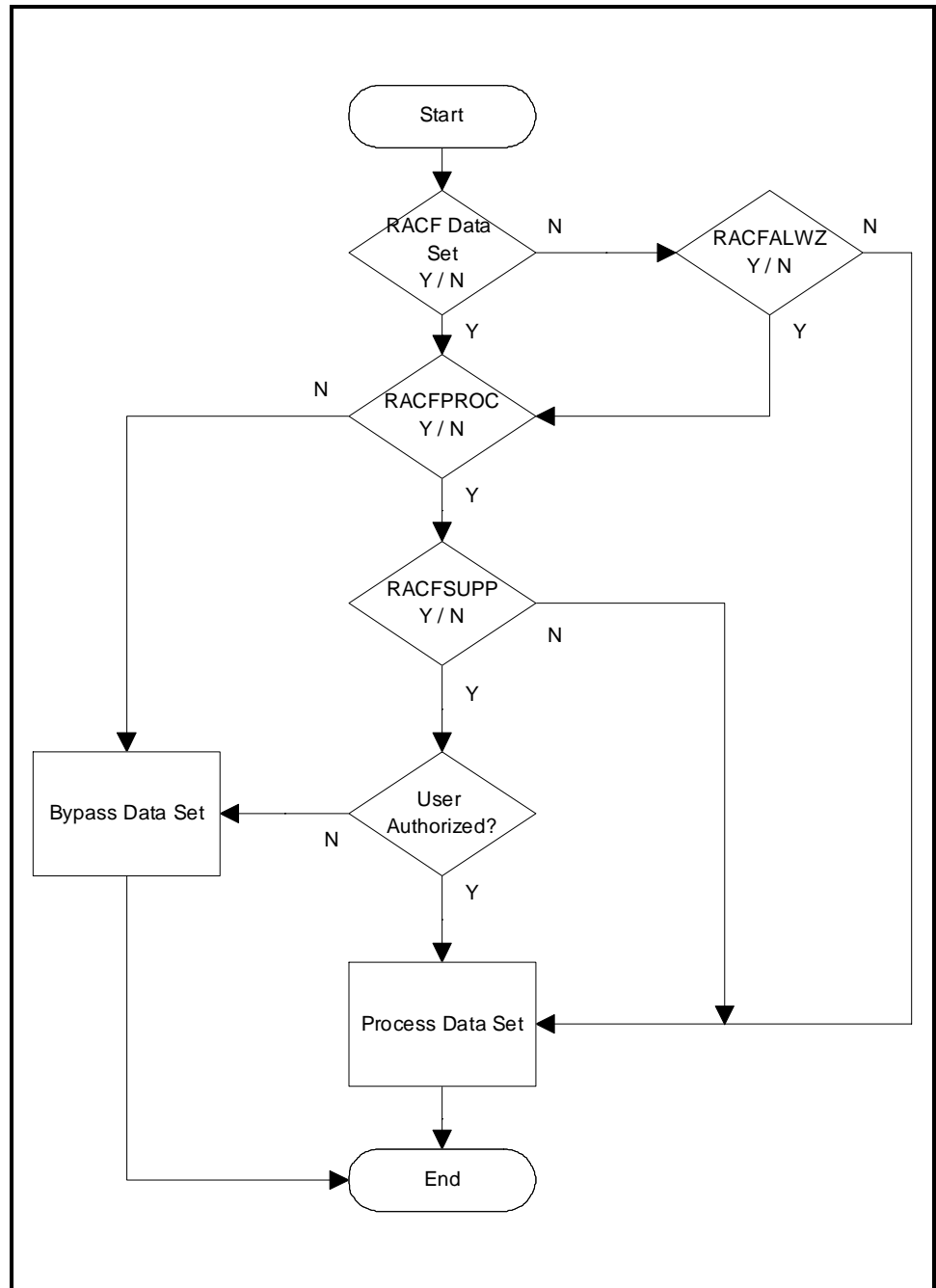


Figure 10-2. RACF Checking Logic Summary Flowchart

Use of RACF Macros

SAMS:Disk interfaces to RACF entirely through standard RACF macro interfaces. The macros used and their purposes are:

RACROUTE REQUEST=AUTH (previously RACHECK)

To determine authorization to process data sets and the existence of generic profiles.

ICHEINTY

To determine the existence of discrete profiles.

RACROUTE REQUEST=DEFINE,TYPE=DEFINE

To copy profiles in Backup, Archive, Restore, Recover and Move/Copy. The MENTITY and MVOLSER fields are used to create the new profile with all profile information copied from the existing profile. All fields are copied across. However, if the userid of the task has the Operations attribute, RACF replaces the OWNER field of the new profile with the userid of the task.

RACROUTE REQUEST=DEFINE,TYPE=ADDVOL

To add a volume to discrete profiles in PDS Compress and Move/Copy.

RACROUTE REQUEST=DELETE,TYPE=ADDVOL

To delete discrete profiles.

RACSTAT

To determine if RACF is active on the system.

RACROUTE REQUEST=VERIFY

To give a started task the same authority as the originating task.

RACROUTE

To invoke the System Authorization Facility (SSF) MSP router.

RACF User Exits

If SAMS:Disk RACF checking is not adequate for users, exit routines have been provided at various points to give user control to validate, reject, modify or monitor SAMS:Disk processing regarding RACF. Following is a list of these SAMS:Disk RACF User Exits:

Table 10-1. List of RACF User Exits

Exit Name	Page #	Brief Description
ICHEKAEX	244	Profile Type Pre-Determination Exit
ICHEKPEX	245	Profile Type Post-Determination Exit
SECURAEX	278	Security Validation (post decision) Exit
SECURPEX	279	Security Validation (pre decision) Exit
RAADDAEX	254	RACF RACDEF-Addvol Post Processing Exit
RAADDPEX	255	RACF RACDEF-Addvol (Pre-RACHECK) Exit
RACFDAEX	256	Post-Decode of SAMS:Disk RACF Profile
RACFDPEX	257	Pre-Decode of SAMS:Disk RACF Profile
RACFEAEX	259	Post-Encode of SAMS:Disk RACF Profile
RACFEPEX	259	Pre-Encode of SAMS:Disk RACF Profile
RACFSMEX	260	Process RACF-Protected DSN with Seq. Migrat
RACHKAEX	261	Post-RACF RACHECK Processing Exit
RACHKPEX	262	Pre-RACF RACHECK Processing Exit
RADEFAEX	263	RACF RACDEF Processing Post-RACDEF Exit
RADEFPEX	264	RACF RACDEF Processing Pre-RACDEF Exit
RADELAEX	265	Post-RACF RACDEF (Delete) Processing Exit
RADELPEX	266	Pre-RACF RACDEF (Delete) Processing Exit

RACF Installation Exits

Since SAMS:Disk uses standard RACF macros to perform RACF profile maintenance, any RACF user exits implemented will receive control for SAMS:Disk functions as well as standard data set processing. This adds to the compatibility of SAMS:Disk with the user's RACF environment. If RACF user exits are implemented, verify that their being invoked by SAMS:Disk will not cause problems. The following two situations should be reviewed:

RACF May Need to Bypass SAMS:Disk Data Set Names

SAMS:Disk profile data set (RESOURCE) names have a special encoded format. They have a standard profile format but user exit routines will not be able to recognize the names. Either they should be bypassed by RACF exit routines or special code may be needed to process them. Modules have been provided to decode and re-encode data set names. They are described below under the heading RACF Internals.

Special RACF Processing for GDG Data Sets

Some users have set up RACF to maintain only a single RACF profile for each GDG base data set name. This improves RACF performance by reducing the number of updates and accesses to the RACF data set. It also simplifies maintenance of GDG profiles by greatly reducing the number of profiles maintained.

SAMS:Disk honors this type of data set protection. While SAMS:Disk does detect the existence of model GDG profiles, SAMS:Disk does not make copies of them in archive, restore or any other function. If your shop uses model GDG profiles, examine closely the sysparm RACGDGVL.

DSNINDEX RACF Flags

The SAMS:Disk DSNINDEX records for the SAMS:Disk archives contain flag bits relating to RACF protection. These flags are reported on the LISTD report. For more detail, turn to the topic "*LISTD Flags*" on page [302](#) of the *User's Guide*.

Format of SAMS:Disk-Saved Profiles

SAMS:Disk optionally copies discrete profiles of data sets that are backed up or archived. This profile is used to re-create profile information when the data set is restored. Since the original data set name cannot be used for the RACF profile name (it may still be in use by the version of the data set still on DASD), and to allow for multiple versions of a profile for a data set, SAMS:Disk creates a special profile name for the data set.

The special profile name is in encoded format. There is an old format and a new format. The old format is referred to as ENCODE FORMAT 1, the new as ENCODE FORMAT 2. In the original SAMS:Disk RACF support, only the first version existed. This version encodes the data set in a binary format that is unreadable and unmaintainable by standard RACF utilities. For this reason, ENCODE FORMAT 2 was developed. It provides the same facilities as the first version but is readable and compatible with all RACF utilities.

A separate SAMS:Disk subfile was created to support the ENCODE TYPE 2 technique (the RACFENCD subfile of the files data set). This file cross-references the archive data set name, date and time to the SAMS:Disk profile name. This file of information is used to find the SAMS:Disk profile name when the data set is to be restored with its discrete profile.

Following are descriptions of the two encode formats:

RACF Encode Profile (Format 1)

If you are currently using this format, we recommend that you convert to Format 2. Format 1 will not support generic profiles. Use the conversion utility documented in this section.

Format 1 is no longer documented, and support for it has been dropped.

RACF Encode Profile (Format 2)

The SAMS:Disk profile name is generated by taking the user-specified prefix, JOBNAME, archive date and archive time, and combining them to make one data set name. The following is the format of the data set name.

```
DSN=uuuuuuu.jjjjjjjj.Dyyddd.THMhhmm.TSSssth
```

uuuuuuuu

The user-specified prefix from sysparm RACFUSID.

jjjjjjjj

The name of the job that caused the data set to be archived.

Dyyddd

D followed by the Julian date on which the data set was archived originally.

THMhhmm

THM followed by the hours and minutes value corresponding to the time at which the data set was archived.

TSSssth

TSS followed by the seconds, tenths and hundredths of seconds values corresponding to the time at which the data set was archived.

Two SAMS:Disk modules perform the encoding and decoding of SAMS:Disk profile data set names. They are distributed in the SAMS:Disk source library and are available for use by user programs if desired. Their program and source member names are:

```
ADSDS019 - ENCODE SAMS:Disk-PROFILE data set name
ADSDS025 - DECODE SAMS:Disk-PROFILE data set name
```

Comments at the beginning of the module describe how to use the modules.

Conversion of SAMS:Disk RACF Profile Encoding

SAMS:Disk can only operate with one profile encoding technique in effect at any one time. If a user wishes to change encoding techniques and currently has SAMS:Disk RACF profiles existing for archived data sets, either all RACF-protected archive data sets must be purged from the archives or a conversion utility must be run to convert existing profiles to the desired format.

Execute the following JCL to convert SAMS:Disk RACF profiles from Format 1 to Format 2:

```
//UNLOAD EXEC PGM=ADSMI002,PARM=ADSDM177
//STEPLIB DD DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//MSGPRINT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//FILES DD DSN=SAMS.DISK.FILES,DISP=SHR
//PARMLIB DD DSN=SAMS.DISK.PARMLIB,DISP=SHR
//SEQFILES DD DISP=(NEW,PASS),DSN=&&DSINDEX,
//          DCB=(LRECL=259,BLKSIZE=6144,DSORG=PS,RECFM=VB),
//          SPACE=(6144,(360,360),RLSE),UNIT=SYSDA
//SYSIN DD *
          UNLOAD FILES=DSNINDEX
//CONVERT EXEC PGM=ADSMI002,PARM='ADSDS058'
//STEPLIB DD DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD DUMMY
//MSGPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//PARMLIB DD DSN=SAMS.DISK.PARMLIB,DISP=SHR
//FILES DD DSN=SAMS.DISK.FILES,DISP=SHR
//SYSUT1 DD DSN=&&DSINDEX,DISP=(OLD,DELETE)
```

Figure 10-3. JCL to Convert SAMS:Disk RACF Profiles

Management of SAMS:Disk-Saved Profiles

Nearly all management of SAMS:Disk-saved profiles is done automatically by the relevant functions involved — Archive/Backup, Restore and IXMAINT. In the event of special cases, four independent utility programs are supplied. They provide the capability to list and maintain RACF profiles. They function on both SAMS:Disk and standard data set profiles.

The three utilities and their functions are:

1. SAMS:Disk-Saved Profile List Utility 1

This utility allows the user to list both SAMS:Disk ENCODED PROFILES and standard data set profiles. The listing format is the

same as would be received from the use of the RACF TSS command LISTDSD.

This utility is provided primarily for those users who have implemented SAMS:Disk PROFILE ENCODE TECHNIQUE 1 (sysparm RACFENTRY1). It provides a technique to display the SAMS:Disk data set name for an otherwise unreadable RACF profile name. Since ENCODE TECHNIQUE 2 profiles are in a standard character format, they can be read directly from the standard RACF profile utility listing.

2. SAMS:Disk-Saved Profile List Utility 2

This utility applies to only those RACF users who have implemented SAMS:Disk RACF ENCODE TECHNIQUE 2 (sysparm RACFENTRY2). It provides a cross-reference listing for SAMS:Disk archived data set names to RACF profile data set names. It provides two reports, the first in sequence by RACF profile data set name, the second in sequence by archive data set name, date and time.

3. SAMS:Disk-Saved Profile Maintenance Utility

This utility allows the user to LIST, DELETE, CREATE and REPLACE RACF profiles. It is designed to be used as a general profile maintenance utility. It provides the capability to process SAMS:Disk ENCODED PROFILES.

SAMS:Disk-Saved Profile List Utility 1

This utility provides the SAMS:Disk RACF user with the capability to list both SAMS:Disk ENCODED PROFILE, and STANDARD DATA SET PROFILE information. It provides a list that is very similar in format to the one received by using the RACF TSS Command LISTDSD. The only difference is the additional information needed to describe SAMS:Disk archive-related data. This information consists of the DATE and TIME the data set was archived. It appears on a line immediately following the Data Set Name line for each profile.

The program runs as a batch program. Following is example JCL for executing the function:

```
// EXEC DMSLSD
//TSS.SYSTSIN DD *
  (standard TSS LISTDSD command goes here)
/*
//DMS.SYSIN DD *
  (DMS SELECT command goes here)
/*
```

Figure 10-4. JCL to Execute Profile List Utility 1

DD Statement Explanations:

SYSTSIN

This input stream must contain a RACF LISTDSD command as input. This command is used to extract the information to be listed from the RACF profile data set. Any options or parameters available for the RACF LISTDSD command may be included to restrict listing output.

To obtain a list of all RACF profiles, use the following command as an example:

```
LISTDSD ALL
```

When a list of only SAMS:Disk profiles is desired, use a command similar to the following:

```
LISTDSD PREFIX(.....) ALL
```

An explanation of the LISTDSD command and its parameters can be found in the RACF Command Language Reference Manual.

SYSIN

This input stream is to contain the SAMS:Disk SELECT command. Its purpose is to further restrict program output to the information specified by the user.

Select Command

The SELECT command restricts list utility output, based on SAMS:Disk profile, DSNAME, or DSNAME PREFIX. If no SELECT command is supplied, all profiles will be listed. Command format and parameter descriptions follow:

```
SELECT DMSONLY,DSNAME=,PREFIX=
```

DMSONLY

This parameter indicates to restrict output to SAMS:Disk profile information only.

DSNAME=

Limit profiles listed to the DSNAME or DSNAME PREFIX specified by this parameter. A maximum of 20 DSNAMEs or DSNAME PREFIXES may be supplied.

PREFIX=

This parameter applies to SAMS:Disk profiles to be listed. It allows limiting the list to those profiles with the 1- to 8-character DSNNAME PREFIX specified. This prefix corresponds to the prefix specified by sysparm RACFUSID.

Normally this parameter is not used because users have only one RACFUSID value specified for all SAMS:Disk profiles.

The following is an example execution of the RACF profile list utility:

```
// EXEC DMSLDSD
//SYSTSIN DD *
LISTDSD ALL
/*
//SYSIN DD *
SELECT DSN=(SYS1./,USER./)
/*
```

Figure 10-5. Example RACF Profile List Utility 1

SAMS:Disk-Saved Profile List Utility 2

This utility creates two cross-reference reports of SAMS:Disk RACF profile data set names to SAMS:Disk archive data set names. One report is in sequence by archive data set name, the other report is in sequence by SAMS:Disk RACF profile data set name.

The program runs as a batch program. Following is example JCL for executing the function:

1988.007 JAN 07, 1988 P R O F I L E X - R E F PAGE 1									
THURSDAY 3.49 PM ARCHIVE DSNAME/DATE/TIME SEQUENCE SAMS:Disk 8.2									
ARCHIVE		ARCHIVE		PROFILE					
DATA SET NAME		TIME	DATE	DATA SET NAME					
<hr/>									
LABS.EDP.CNTL		0758	1986.212	SBEDP.AUTOBUP.D86212.THM0758.TSS5923					
LABS.EDP.CNTL		1722	1987.002	SBEDP.AUTOBUP.D87002.THM1722.TSS2005					
LABS.EDP.CNTL		1645	1987.005	SBEDP.AUTOBUP.D87005.THM1645.TSS1033					
LABS.EDP.CNTL		1731	1987.006	SBEDP.AUTOBUP.D87006.THM1731.TSS5063					
LABS.EDP.CNTL		1723	1987.007	SBEDP.AUTOBUP.D87007.THM1723.TSS2577					
LABS.EDP.P\$D		1720	1987.002	SBEDP.AUTOBUP.D87002.THM1720.TSS0507					
LABS.EDP.P\$D		1620	1987.005	SBEDP.AUTOBUP.D87005.THM1620.TSS4007					
LABS.EDP.P\$GD		1720	1987.002	SBEDP.AUTOBUP.D87002.THM1720.TSS2471					
LABS.EDP.P\$GD		1622	1987.005	SBEDP.AUTOBUP.D87005.THM1622.TSS3506					
<hr/>									
TOTAL DATA SETS			9						

Figure 10-6. Example JCL to Execute Profile Utility 2

DD Statement Explanations:

SYSIN

This input stream is to contain the SAMS:Disk SELECT command. Its purpose is to further restrict program output to the information specified by the user.

Select Command

The SELECT command restricts list utility output, based on DSNAME or DSNAME PREFIX. If no parameters are specified with the SELECT command, all profiles will be listed. Command format and parameter description follow:

```
SELECT DSNAME=, PROFILE=
```

DSNAME=

Limit archive data set names listed to the DSNAME or DSNAME PREFIX specified by this parameter. A maximum of 20 DSNAMEs or DSNAME PREFIXES may be supplied.

PROFILE=

Limit profile names listed to the NAME or NAME PREFIX specified by this parameter. A maximum of 20 names or prefixes may be supplied.

Examples

The following is an example execution of the RACF PROFILE LIST UTILITY 2. Running this utility generates both reports.

```

1988.007 JAN 07, 1988 P R O F I L E X - R E F PAGE 1
THURSDAY 3.49 PM PROFILE NAME SEQUENCE SAMS:Disk 8.2

PROFILE ARCHIVE ARCHIVE
DATA SET NAME DATA SET NAME TIME DATE

SBEDP.AUTOBUP.D86212.THM0758.TSS5923 LABS.EDP.CNTL 0758 1986.212
SBEDP.AUTOBUP.D87002.THM1720.TSS0507 LABS.EDP.P$D 1720 1987.002
SBEDP.AUTOBUP.D87002.THM1720.TSS2471 LABS.EDP.P$GD 1720 1987.002
SBEDP.AUTOBUP.D87002.THM1722.TSS2005 LABS.EDP.CNTL 1722 1987.002
SBEDP.AUTOBUP.D87006.THM1731.TSS5063 LABS.EDP.CNTL 1731 1987.006
SBEDP.AUTOBUP.D87007.THM1723.TSS2577 LABS.EDP.CNTL 1723 1987.007
SBEDP.AUTOBUP.D87005.THM1620.TSS4007 LABS.EDP.P$D 1620 1987.005
SBEDP.AUTOBUP.D87005.THM1622.TSS3506 LABS.EDP.P$GD 1622 1987.005
SBEDP.AUTOBUP.D87005.THM1645.TSS1033 LABS.EDP.CNTL 1645 1987.005

TOTAL DATA SETS 9

```

Figure 10-7. Example Execution of the RACF Utility 2

RACF Profile to Archive Dsname X-REF Reports

The following is a sample of the Profile X-REF report.

```

1988.007 JAN 07, 1988 P R O F I L E X - R E F PAGE 1
THURSDAY 3.49 PM ARCHIVE DSNAME/DATE/TIME SEQUENCE SAMS:Disk 8.2

ARCHIVE ARCHIVE PROFILE
DATA SET NAME TIME DATE DATA SET NAME

LABS.EDP.CNTL 0758 1986.212 SBEDP.AUTOBUP.D86212.THM0758.TSS5923
LABS.EDP.CNTL 1722 1987.002 SBEDP.AUTOBUP.D87002.THM1722.TSS2005
LABS.EDP.CNTL 1645 1987.005 SBEDP.AUTOBUP.D87005.THM1645.TSS1033
LABS.EDP.CNTL 1731 1987.006 SBEDP.AUTOBUP.D87006.THM1731.TSS5063
LABS.EDP.CNTL 1723 1987.007 SBEDP.AUTOBUP.D87007.THM1723.TSS2577
LABS.EDP.P$D 1720 1987.002 SBEDP.AUTOBUP.D87002.THM1720.TSS0507
LABS.EDP.P$D 1620 1987.005 SBEDP.AUTOBUP.D87005.THM1620.TSS4007
LABS.EDP.P$GD 1720 1987.002 SBEDP.AUTOBUP.D87002.THM1720.TSS2471
LABS.EDP.P$GD 1622 1987.005 SBEDP.AUTOBUP.D87005.THM1622.TSS3506

TOTAL DATA SETS 9

```

Figure 10-8. Sample Profile X-REF Report

Note: This sample report is in dsname/data/time sequence and has had the data set names shortened for printing on this page.

RACF Profile to Archive Dsname X-REF Reports

The following is a sample of the Profile X-REF report.

1988.007	JAN 07, 1988	P R O F I L E	X - R E F	PAGE	1
THURSDAY 3.49 PM		PROFILE NAME	SEQUENCE	SAMS:Disk	8.2
PROFILE		ARCHIVE		ARCHIVE	
DATA SET NAME		DATA SET NAME	TIME	DATE	
<hr/>					
SBEDP.AUTOBUP.D86212.THM0758.TSS5923	LABS.EDP.CNTL	0758	1986.212		
SBEDP.AUTOBUP.D87002.THM1720.TSS0507	LABS.EDP.P\$D	1720	1987.002		
SBEDP.AUTOBUP.D87002.THM1720.TSS2471	LABS.EDP.P\$GD	1720	1987.002		
SBEDP.AUTOBUP.D87002.THM1722.TSS2005	LABS.EDP.CNTL	1722	1987.002		
SBEDP.AUTOBUP.D87006.THM1731.TSS5063	LABS.EDP.CNTL	1731	1987.006		
SBEDP.AUTOBUP.D87007.THM1723.TSS2577	LABS.EDP.CNTL	1723	1987.007		
SBEDP.AUTOBUP.D87005.THM1620.TSS4007	LABS.EDP.P\$D	1620	1987.005		
SBEDP.AUTOBUP.D87005.THM1622.TSS3506	LABS.EDP.P\$GD	1622	1987.005		
SBEDP.AUTOBUP.D87005.THM1645.TSS1033	LABS.EDP.CNTL	1645	1987.005		
<hr/>					
TOTAL DATA SETS	9				

Figure 10-9. Sample X-REF Report by Profile Name

Note: This sample report is in profile name sequence and has had the data set names shortened for printing on this page.

SAMS:Disk-Saved Profile Maintenance Utility

This utility provides a tool to maintain SAMS:Disk profile information. It gives the capability to create, delete, list and rename RACF profile data. A user must be authorized to update the profile of a data set before using this program to maintain it.

The utility should be used to maintain discrete profiles only. Do not attempt to do generic profile maintenance with it.

The program is executed in a batch environment. Transaction listings are created to indicate the status of the requests entered.

Based on input command parameters, this utility can be used to maintain both SAMS:Disk profiles and standard data set profiles (discrete profiles only). Multiple input commands may be supplied to process different profiles.

Four different commands are accepted by the utility:

1. **DISPLAY** — This command requests that the profile data for a specific RACF profile be displayed. The following fields are shown:

data set name
volume(s)
data set type
universal access
auditing
owner ID
installation data

access list — group name and access level

2. **DELETE** — Delete the specified RACF profile. Take no action on the data set associated with the profile.
3. **CREATE** — This command allows a new profile to be generated from an existing one. The user specifies the name and volume of the existing profile to be used as a model to create the new one.
4. **RENAME** — Assign a new name and volume to a RACF profile. If the profile is associated with an existing data set, the original data set will no longer have a profile.

Execution JCL Example

```
// EXEC  DMSPROF
//SYSIN  DD  *
(maintenance commands go here)
/*
```

Display Command

```
DISPLAY DSNAME=,VERSION=,NONDMS,VOLUMES=,VSAM
```

DSNAME=

Specifies the 1- to 44-character data set name of the profile to be listed. This must be a complete name. It may contain a generation data set version number if desired.

If this request is to display a SAMS:Disk profile, a date and time value must be used to determine the encoded data set name for the profile. Both a date and time command parameter may accompany the data set name to be used to find the profile. If these are not supplied, the DSNINDEX record will be used to determine the archive date and time for the data set. The date and time parameters are not normally supplied with the command unless the DSNINDEX record is not available or is unusable.

VERSION=

If the DSNINDEX file is to be used to determine the date and time for a SAMS:Disk profile, this parameter may indicate the version of a data set in the index if multiple copies have been archived (that is, VERSION=(-01)).

VOLUMES=

If the request is for a SAMS:Disk profile, this parameter indicates the volume with which SAMS:Disk has the profile associated. In this case, the VOLUME parameter is not normally supplied because SAMS:Disk always associates profiles with the volume specified on sysparm RACFDVOL. The value from the sysparm is used as a default for this parameter when SAMS:Disk profiles are being processed.

If this request is for a standard data set profile, SAMS:Disk needs the volumes on which the data set is protected. If this parameter is not specified, the system catalog will be used to determine the volume serials. If the data set is not cataloged, specify one or more volumes on which this data set is protected.

NONDMS

This simple parameter indicates the data set name supplied is for a standard data set profile. NOTE: DATE and TIME parameters are not valid when this parameter is specified.

If this parameter is not specified, it is assumed the request is for a SAMS:Disk profile.

VSAM

Specify this parameter to indicate the request is for a profile associated with a VSAM data set. When not specified, the request is assumed to be for a non-VSAM data set.

Delete Command

```
DELETE DSN= , VERSION= , NONDMS , VOLUMES= , VSAM , LIST
```

DSNAME=

(See *DSNAME* parameter on page 411 in this manual.)

VERSION=

(See *VERSION* parameter on page 411 in this manual.)

VOLUMES=

(See *VOLUMES* parameter on page 411 in this manual.)

NONDMS

(See *NONDMS* parameter on page 412 in this manual.)

VSAM

(See *VSAM* parameter on page 412 in this manual.)

LIST

This parameter causes the profile being deleted to be displayed before deletion. The format in which it is displayed is the same format as the one generated by the DISPLAY command.

Create Command

```
CREATE OLDDSN= , OLDVER= , OLDVOLS= , NEWDSN= , NEWVER= , NEWVOLS= ,  
      OLDNONDMS , NEWNONDMS , LIST , VSAM
```

OLDDSN=

(See *DSNAME* parameter on page 411 in this manual.)

OLDVER=

Pertains to the model (original) profile being used. (See *VERSION* parameter on page 411 in this manual.)

OLDVOLS=

This parameter indicates the volume(s) for which the model profile is defined. (See *VOLUME* parameter on page 411 in this manual.)

NEWDSN=

Specifies the 1- to 44-character data set name of the new profile to be created. (See *DSNAME* parameter on page 411 in this manual.)

NEWVER=

Pertains to the new profile. (See *VERSION* parameter on page 411 in this manual.)

NEWVOLS=

This parameter indicates the volume(s) to be assigned to the new profile. (See *VOLUME* parameter on page 411 in this manual.)

OLDNONDMS

This simple parameter indicates the data set name supplied for the model (original) profile is not for a SAMS:Disk profile.

NEWNONDMS

Same meaning and format as the *OLDNONDMS* parameter, except that it applies to the profile to be created.

VSAM

Specify this parameter to indicate the request is for a profile associated with a VSAM data set. When not specified, the request is assumed to be for a non-VSAM data set.

LIST

This parameter causes the profile created to be displayed. The format in which it is displayed is the same format as the one generated by the *DISPLAY* command.

Rename Command

```
RENAME OLDDSN= , OLDVER= , OLDDVOLS= , NEWDSN= , NEWVER= , OLDNONDMS ,  
NEWNONDMS , LIST , VSAM
```

OLDDSN=

(See *DSNAME* parameter on page 411 in this manual.)

OLDVER=

Pertains to the model (original) profile being used. (See *VERSION* parameter on page 411 in this manual.)

OLDVOLS=

This parameter indicates the volume(s) on which the profile being renamed is defined. (See *VOLUME* parameter on page 411 in this manual.)

NEWDSN=

Specifies the 1- to 44-character data set name to be assigned to the profile. (See *DSNAME* parameter on page 411 in this manual.)

NEWVER=

Pertains to the new profile. (See *VERSION* parameter on page 411 in this manual.)

OLDNONDMS

This simple parameter indicates the profile being renamed is not a SAMS:Disk profile.

NEWNONDMS

Same meaning and format as the OLDNONDMS parameter except that it applies to the new profile name.

VSAM

Specify this parameter to indicate the request is for a profile associated with a VSAM data set. When not specified, the request is assumed to be for non-VSAM.

LIST

This parameter causes the profile to be displayed. The format in which it is displayed is the same format as the one generated by the DISPLAY command.

Execution Examples:

The following example demonstrates displaying the RACF profile information for a SAMS:Disk RACF profile:

```
// EXEC  DMSPROF
//SYSIN DD  *
DISPLAY DSNAME=USER.LOAD.LIBRARY
/*
```

The following example shows the deletion of the same SAMS:Disk RACF profile:

```
// EXEC  DMSPROF
//SYSIN DD  *
```

```
DELETE DSN=USER.LOAD.LIBRARY
/*
```

The following example shows the deletion of a non-SAMS:Disk RACF profile:

```
// EXEC DMSPROF
//SYSIN DD *
DELETE DSN=USER.PROD.LIBRARY,NONDMS
/*
```

The following example shows the creation of a profile for DSN=USER.DSN1 from an already existing profile USER.TEST2. Both data sets are on DASD and cataloged.

```
// EXEC DMSPROF
//SYSIN DD *
CREATE NEWDSN=USER.DSN1,NEWNONDMS,OLDDSN=USER.TEST2,OLDNONDMS
/*
```

Changing the Value of RACFDVOL

The following procedure describes how to change the value of RACFDVOL. This sysparm contains the real online DASD volume name that is required by RACF for SAMS:Disk-saved profiles. Actually, the steps describe what to do before you change the sysparm. Changing the sysparm is quite easy, but without doing the preparatory steps described here, system abends will occur. The jobs are long running, so submit them in batch mode.

1. Run the following TSS command in batch. Use the current values for sysparms RACFUSID and RACFDVOL, and the desired value of the sysparm RACFDVOL:

```
SEARCH MASK(racfusid.,THM) VOLUME(olddvol)
NOLIST NOGENERIC + CLIST('ALTDSD ','ALTVOL(olddvol newdvol)')
```

The TSS command creates a CLIST in the data set 'userid.EXEC.RACF.CLIST'. This data set contains TSS commands. There is one TSS command for each SAMS:Disk-saved RACF profile that will change its associated volume from the old DVOL to the new DVOL.

2. Verify that the CLIST contains a command for every SAMS:Disk-saved profile, and no others.
3. Do step 1 until you are sure that the list contains every SAMS:Disk-saved profile.
4. Plan a time and procedure for steps 5 through 10.
5. Make sure that no SAMS:Disk Backup, Archive or Restore jobs are running.

6. Backup the SAMS:Disk Files data set and the RACF data base.
7. Do step 1 again.
8. Use batch mode to execute the CLIST you just created in 'youruserid.EXEC.RACF.CLIST'.
9. Verify that each command in the CLIST completed successfully.
10. Change the setting of SAMS:Disk sysparm RACFDVOL to the new value. The change is now complete, and you can allow SAMS:Disk processing again.

Chapter 11. Report Processing

SAMS:Disk report processing is accomplished by generating report detail records, sorting them, and passing them to a common report writer routine. Each report detail record is prefixed by its 10-byte report table name and its report sequencing control field. The sequencing control field may optionally be used to provide page control breaks for the report. Only one control field may be used to generate page control breaks.

SAMS:Disk places report detail records for all reports requested into one common output data set. After all detail report records are generated, a sort step is executed to order the records by report table name/control field sequence. These records are then passed to the common report writer routine.

The report writer uses the report table name to retrieve the proper table to use to format the report. The format control table contains all information needed to build the printed report line from the supplied detail records. The contents and field definitions for the formatting control tables is documented later in this section. When a detail record is read for a different report, the new format control table is loaded and the processing continues. A flow diagram describing this process is also presented in a later section.

REPORTNAME Table — Member REPORTNM

The report name table (member REPORTNM in the parmlib data set) relates modules that construct output records to the tables used by the report writer to format the respective reports. The content of the entries in the report name table is identified below.

Table 11-1. REPORTNM Content Descriptions

Cols	Description		
1-8	Name of module that produces report output records		
9	Character indicating which type of DSCB during VTOC SCAN is to be passed to the module:	1	Format-1 DSCB only.
		5	Format-5 DSCB only. - All DSCB types when blank.
		V	Do not pass any DSCB - Invoke at end of volume for extraction of summary data.
10-19	10-character name of the report table to be used by the report writer in formatting the report		

The last six characters of each report table name serve as the name of the respective report for the REPORT and DISPLAY commands.

REPORTNAME Table Example:

TE = 'ADSDM0521RPTNMVDICT'

Table 11-2. REPORTNM Example

Value	Definition
ADSDM052	Name of module to be given control to produce the Multiple Volume Dictionary Report records
1	Format-1 DSCBs are the only ones passed to the report module
RPTNMVDICT	Table used to format the Multiple Volume Dictionary Report

Format-1 / Format-4 DSCB Appendage Fields

Allocation information regarding individual data sets that is not available in the format-1 DSCB is maintained as an appendage to the DSCB, which is passed to the modules that assemble the data for the report record. Allocation information that has been accumulated from the processing of all DSCB types is available as an appendage to the format-4 DSCB, which is also passed as one of the parameters to the report assemble modules. The fields are described in the appendix of this manual.

Tables for Report Formatting

The majority of the SAMS:Disk reports are formatted by the use of tables that are stored in the parmlib data set. Any change made to a table will be reflected the next time the subject report is printed, since tables are loaded at execution time.

Four types of table entries, or records, are used in report preparation. These are the CONTROL, HEADER, DETAIL, and TOTAL types that are described below.

1. Control Entry

Only one occurrence of this type may be present for each report format control table. It contains control information about the layout of a specific report.

Table 11-3. Control Entry Description

Cols	Description
1	'C' for entry type
2-4	(not used)
5-7	Limit number of lines per page for overflow. (Overridden by RPTLINES sysparm).
8-10	Rows of detail lines per page [*]
11-13	Columns of detail per page [*]
14-16	Length of print detail to be replicated [*]
17-19	Length of source record for each detail set [*]
20-22	Length of control field that will force page overflow (control breaks)

^{*} Required for N-up reports.

2. Header Entry

An entry of this type is required for each field and literal to be printed on a report heading line.

Table 11-4. Header Entry Description

Cols	Description
1	'H' for entry type

Cols	Description
2	The character placed in this position is the identifier for all entries comprising the given heading line
3	The carriage control character for the heading line in question must be specified in this position for only the first entry of each line
4	Field type code
7	Length of the source field in bytes
10	Position of field in source record — left blank for literals
11-13	Position on print line for field placement
14-16	Target length of field where different from source length (halfword to zoned)
17-nn	Field type dependent

3. Detail Entry

An entry of this type is required for each field and literal to be placed on a report detail line.

Table 11-5. Detail Entry Description

Cols	Description
1	'D' for entry type
2	'1' is normally placed in the identifier field since only one type of detail line is presently supported
3	The carriage control character to be used for all detail lines is specified here
4	Field type code
7	Length of the source field in bytes
10	Position of field in source record — left blank for literals
11-13	Position on print line for field placement
14-16	Target length of field where different from source length (halfword to zoned)
17-nn	Field type dependent

Fields and literals are specified in the same manner for both of the applicable line types.

Table 11-6. Possible Line Type Values and Descriptions

Cols	
4	Field type code
7	Length of the source field in bytes
10	Position of field in source record — left blank for literals
11-13	Position on print line for field placement
14-16	Target length of field where different from source length (halfword to zoned)
17-nn	Field type dependent

4. Total Entry

An entry of this type is required for each field and literal to be placed on a report total line.

Table 11-7. Total Entry Descriptions

Cols	Description
1	'T' for entry type
2	Numeric indicator of total line occurrence to which this entry applies
3	The carriage control character to be used for all total lines is specified here
4	Field type code
5-7	Length of the source field that is to be accumulated. The source field must be either a halfword or fullword in the record. If this value is zero, a length of 2 is assumed. Valid values are 000, 002 or 004.
8-10	Position of field in source record — left blank for literals. If this value is zero, the accumulator is assumed to be a count of records processed.
11-nn	All remaining fields have the same meanings as for detail type records. (See the description of the detail entry for their formats and use.)

Described below are the various field types and special considerations for their use (position 4 in table entry).

Table 11-8. Special Field Types for Position 4

Value	Description
A	Accumulator for Report Totals
B	Binary
C	Character
D	Date
E	Security
F	Fullword
H	Halfword
I	Installation Name
J	Create/Expire Date Expanded Form
K	Constant / Literal
M	Mask
N	Page Number
P	Packed Decimal
R	Device Type
S	SAMS:Disk System Header
T	Time
U	Allocation Type
V	Data Set Organization
W	Record Format
X	Hexadecimal
Y	Create/Expire Date
Z	Zoned Decimal
1	Mount Status
2	Use Status
3	Standard Header Line 1
4	Standard Header Line 2
\$	Dollars and Cents

Format Table Examples

The following table entry examples are taken from the RPTNOMVDIC table (member RPTNOMVD in the parmlib data set).

Control Entry Example

TE = 'C1 058054001132126020'

Table 11-9. RPTNOMVD Example

Cols	Sample	Description
1		C entry type
4		(not used)
5-7	058	Lines per page
8-10	054	Detail lines per page
11-13	001	Columns of detail
14-16	132	Length of detail line
17-19	126	Length of source record
20-22	020	Length of control field that will force page overflow; in this case, when the 20 characters of the owner name change, the detail record is printed on a new page.

Header Entry Examples

TE = 'H5 K013 006 DATA SET NAME'

Table 11-10. Header Entry Example

Cols	Sample	Description
1	H	Entry type - header
2	5	Line 5 of the header
3		(not used)
4	K	Literal - "DATA SET NAME"
5-7	013	Length of literal
8-10		Blanks - not used with a literal
11-13	006	Position in print line where literal begins
14-16		Blanks - not used with a literal
17-28	DS...ME	The literal value that is placed in the print line

Detail Entry Examples

TE = 'D1 C006071033 VOL'

Table 11-11. Detail Entry Example #1

Cols	Sample	Description
1	D	Entry type - detail
2	1	Line 1 of detail
3		Blank - single space
4	3	Literal in source
5-7	006	Length of literal in source
8-10	071	Position of literal in source relative to table name
11-13	033	Position in print line where the literal is to begin
14-19		Blanks - not used with a literal
20-22	VOL	Comments

TE = 'D1 H002082069005Z BLKSIZE'

Table 11-12. Detail Entry Example #2

Cols	Sample	Description
1	D	Entry type - detail
2	1	Line 1 of detail
3		Blank - single space
4	H	Data in source is a halfword
5-7	002	Length of data in source
8-10	082	Position of data in source relative to the table name
11-13	069	Position in print line where data is to begin
14-16	005	Maximum length of data in print line is five characters
17	Z	Suppress leading zeros
18-19		Blanks - not used
20-26	BLKSIZE	Comment

Total Entry Examples

TE = 'T10K015 005 TOTAL DATA SETS'

Table 11-13. Total Entry Example #1

Cols	Sample	Description
1	T	Entry type - total
2	1	Line 1 of total
3	0	Double space
4	K	Constant total header
5-7	015	Length of constant
11-13	005	Position in print line where the literal is to begin
17-28	xxx	The constant value that is placed in the print line

TE = 'T1 A000000021005Z BLKSIZE'

Table 11-14. Total Entry Example #2

Cols	Sample	Description
1	T	Entry type - total
2	1	Line 1 of total
3		Blank - single space
4	A	This is an accumulator field
5-7	000	Due to the type of accumulator (count), this value is ignored
8-10	000	Accumulator is to count number of records passed to reporting routine
11-13	069	Position in print line where total is to begin
14-16	005	Maximum length of data in print line is five characters
17	Z	Suppress leading zeros
18-19		Blanks - not used
20-26	BLKSIZE	Comment

Note: Each source record begins with two halfwords followed by the name of the table to be used to print the record.

REC	TABLE	Position in SOURCE		
LEN	NAME	1	21	27
01260000	RPTNOMVDIC	OWNER NAME... PROJNO DSN...		

Report Modification Example

If a change were to be made to the Multiple Volume Dictionary Report, such as putting an owner in the heading, the following changes will need to be made.

1. Add an entry to the SYSPARMS member in parmlib.

RPTMVDEX*exitname* - where *exitname* is the name of the module to be given control to write an owner record.

2. Add a new formatting table as a member in parmlib that is to be used for printing the report.
3. Add the following entry to the report name table (member REPORTNM in parmlib).

ADSDM0521RPTNOMVDIC

4. Write an exit module that will call ADSDM033 to write the new record based on the following parameters passed to the exit module:

Table 11-15. Parameters Passed to module ADSDM033

Parameters	Description		
P1	format-4 DSCB and appendage fields volume serial is in positions 1 to 6		
P2	format-1 DSCB and appendage fields		
P3	standard report record for Multiple Volume Dictionary Report		
P4	Halfword return code	1	WRITE standard Multiple Volume Dictionary record
		2	DO NOT WRITE standard MVD record

The report may be invoked using the following command:

```
REPORT RPT=NOMVDIC,VOL=VOL001
```


Example of Owner Table

In this table provision has been made for an OWNER in the header and a PROJECT NUMBER in each detail line.

MEMBER=RPTNOMVD in the parmlib data set

```
'C1 058051001132116020'
'H113132 001 standard header - line 1'
'H1 K007 122 PAGE '
'H1 N004 130'
'H1 K032 032 M U L T I P L E V O L U M E '
'H1 K037 064 D A T A S E T D I C T I O N A R Y'
'H2 4132 001 standard header - line 2'
'H3 K001 001132'
'H4 K005 001 OWNER'
'H4 C020001008 OWNER'
'H5 K011 038 PSWD DEVICE'
'H5 K029 081 ---ALLOCATION--- -TRACKS--- '
'H6 K013 001 DATA SET NAME'
'H6 K035 027 VOLUME CAT PROT TYPE PROJ DSORG'
'H6 K029 063 RECFM BLKSZ LRECL TYPE PRIM'
'H6 K016 094 SEC ALLOC IDLE'
'H6 K021 112 EXT CREDIT EXPDT USED'
'H7 K040 001 _____ -- -- --'
'H7 K040 041 - ____ _ _ _ _ _ _ _ _ '
'H7 K040 081 _____ -- -- --'
'H7 K012 121 ____ _'
'D1 C044027001025 DSN'
'D1 C006071027 VOL'
'D1 C001108035 CAT'
'D1 E001099040 SEC'
'D1 R002077043 DEVICE TYPE'
'D1 C006021050 PROJECT NUMBER'
'D1 V002079059 DSORG'
'D1 W001081064 RECFM'
'D1 H002082069005Z BLKSIZE'
'D1 H002084075005Z LRECL'
'D1 U001086082003 ALLO TYPE'
'D1 H002100086005Z ALLO PRI'
'D1 B003087092005Z ALLO SEC'
'D1 B003090099005Z ALLO SUM'
'D1 H002102105005Z ALLO IDL'
'D1 B001112112002Z EXT NO' .
'D1 Y002093116 CREATE'
'D1 Y002096122 Z EXPIRE'
'D1 Y003109128 Z LAST USED'
'T1 K040 001_____ '
'T1 K040 041 _____ '
'T1 K040 081 _____ '
```

```
'T1 K012 121 -----'
'T2 K015 010 TOTAL DATA SETS'
'T2 A000000026007Z COUNTER'
'T2 K022 072 TOTAL ALLOCATED TRACKS'
'T2 A000091095009Z ACCUMULATOR'
'T3 K017 072 TOTAL IDLE TRACKS'
'T3 A000102095009Z ACCUMULATOR'
```

Compare the RPTNOMVD member table with the standard MVD table below.

Standard Multiple Volume Dictionary Report Table

MEMBER=RPTNMVDI in the parmlib data set.

```
'C1 058052001132090'
'H113132 001'
'H1 K007 122 PAGE '
'H1 N004 130'
'H1 K032 032 M U L T I P L E V O L U M E '
'H1 K037 064 D A T A S E T D I C T I O N A R Y'
'H2 4132 001'
'H3 K001 001132'
'H4 K011 040 PSWD DEVICE'
'H4 K029 081 --ALLOCATION-- -TRACKS-- '
'H5 K013 001 DATA SET NAME'
'H5 K033 029 VOLUME CAT PROT TYPE ADDR DSORG'
'H5 K029 063 RECFM BLKSZ LRECL TYPE PRIM'
'H5 K016 094 SEC ALLOC IDLE'
'H5 K021 112 EXT CREDIT EXPDT USED'
'H6 K040 001 ----- -- -- -- '
'H6 K040 041 -- -- -- -- -- -- -- '
'H6 K040 081 ----- -- -- -- '
'H6 K012 121 -- -- '
'D1 C044001001028 DSN'
'D1 C006045029 VOL'
'D1 C001082037 CAT'
'D1 E001073042 SEC'
'D1 R002051045 DEVICE TYPE'
'D1 C003078052 ADDRESS'
'D1 K001 055 '
'D1 C001081056 SUB-MODULE'
'D1 V002053059 DSORG'
'D1 W001055064 RECFM'
'D1 H002056069005Z BLKSIZE'
'D1 H002058075005Z LRECL'
'D1 U001060082003 ALLO TYPE'
'D1 H002074086005Z ALLO PRI'
'D1 B003061092005Z ALLO SEC'
'D1 B003064099005Z ALLO SUM'
'D1 H002076105005Z ALLO IDL'
'D1 B001086112002Z EXT NO'
```

```

'D1 Y002067116 CREATE'
'D1 Y002070122 Z EXPIRE' . .
'D1 Y003083128 Z LAST USED'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K015 010 TOTAL DATA SETS'
'T2 A000000026007Z COUNTER'
'T2 K022 072 TOTAL ALLOCATED TRACKS'
'T2 A000065095009Z ACCUMULATOR'
'T3 K017 072 TOTAL IDLE TRACKS'
'T3 A000076095009Z ACCUMULATOR'

```

Example of Record Format for Owner Record

The following table describes the record format of the Owner Record:

Table 11-16. Record Format for the Owner Record

Name	Type	Size	Offset	Description
ownrecrd	DS	0F		
	DC	h'126,0'		Record length
owntable	DC	CL10'RPTNBMVDIC'		Report table name
ownid	DC	CL20' '	1	Owner ID
ownprojn	DC	CL6' '	21	Project number
owndsnam	DC	CL44' '	27	Data set name
ownvolno	DC	CL6' '	71	Volume
owndevcl	DC	XL2'0000'	77	Device class
owndsorg	DC	XL2'0000'	79	DSORG
ownrecfm	DC	XL1'00'	81	RECFM
ownblksi	DC	XL2'0000'	82	BLKSIZE
ownlrecl	DC	XL2'0000'	84	LRECL
ownscalo	DC	XL1'00'	86	Allocation (trk,cyl,...)
ownscqty	DC	XL3'000000'	87	Second alloc quantity
owntksum	DC	XL3'000000'	90	Total allocation
owncred	DC	XL3'000000'	93	Creation date

Name	Type	Size	Offset	Description
ownexpdt	DC	XL3'000000'	96	Expiration date
owndsind	DC	XL1'00'	99	Security
ownprmal	DC	XL2'0000'	100	Primary allocation
ownidlet	DC	XL2'0000'	102	Idle
ownvolad	DC	CL4' '	104	Volume address
owncatlg	DC	CL1' '	108	Cataloged
ownlusdt	DC	XL3'000000'	109	Last used date
ownnoext	DC	XL1'00'	112	Number of extents

Compare the OWNER RECORD to the standard MVD record that follows.

Standard Multiple Volume Dictionary Record Format

The following table describes the record format of the Standard Multiple Volume Dictionary:

Table 11-17. Record Format of the Standard MVD

Name	Type	Size	Offset	Description
rptrecrd	DS	Of,		
	DC	h'100,,0'		Record length
rpttable	DC	CL10'REPTNMVDICT'		Report table name
rptdsnam	DC	CL44' '	1	Data set name
rptvolno	DC	CL6' '	45	Volume
rptdevcl	DC	XL2'0000'	51	Device class
rptdsorg	DC	XL2'0000'	53	DSORG
rptrecfm	DC	XL1'00'	55	RECFM
rptblksi	DC	XL2'0000'	56	BLKSIZE
rptlrecl	DC	XL2'0000'	58	LRECL
rptscalo	DC	XL1'00'	60	Allocat (trk,cyl,etc)
rptscqty	DC	XL3'000000'	61	Second alloc quantity

Name	Type	Size	Offset	Description
rpttksum	DC	XL3'000000'	64	Total allocation
rptcred	DC	XL3'000000'	67	Creation date
rptexpdt	DC	XL3'000000'	70	Expiration date
rptdsind	DC	XL1'00'	73	Security
rptprmal	DC	XL2'0000'	74	Primary allocation
rptidlet	DC	XL2'0000'	76	Idle
rptvolad	DC	CL4' '	78	Volume address
rptcatlg	DC	CL1' '	82	Cataloged
rptlusdt	DC	XL3'000000'	83	Last used date
rptnoext	DC	XL1'00'	86	Number of extents

Report Logic Diagram

```
REPORT RPT=(MVD,SVD,DSI),VOL=(VOL001,VOL002)
```

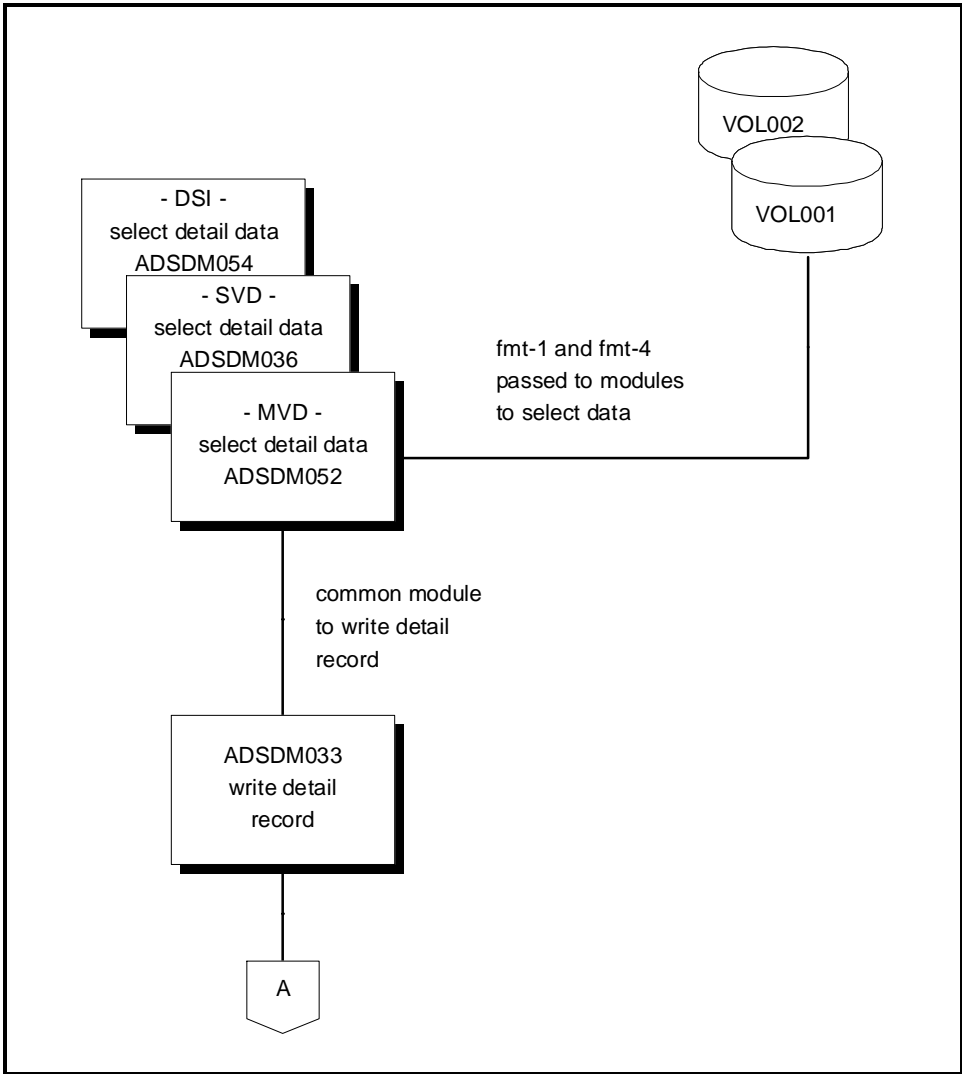


Figure 11-1. Report Logic Diagram

Each record is a detail line in a report as illustrated below:

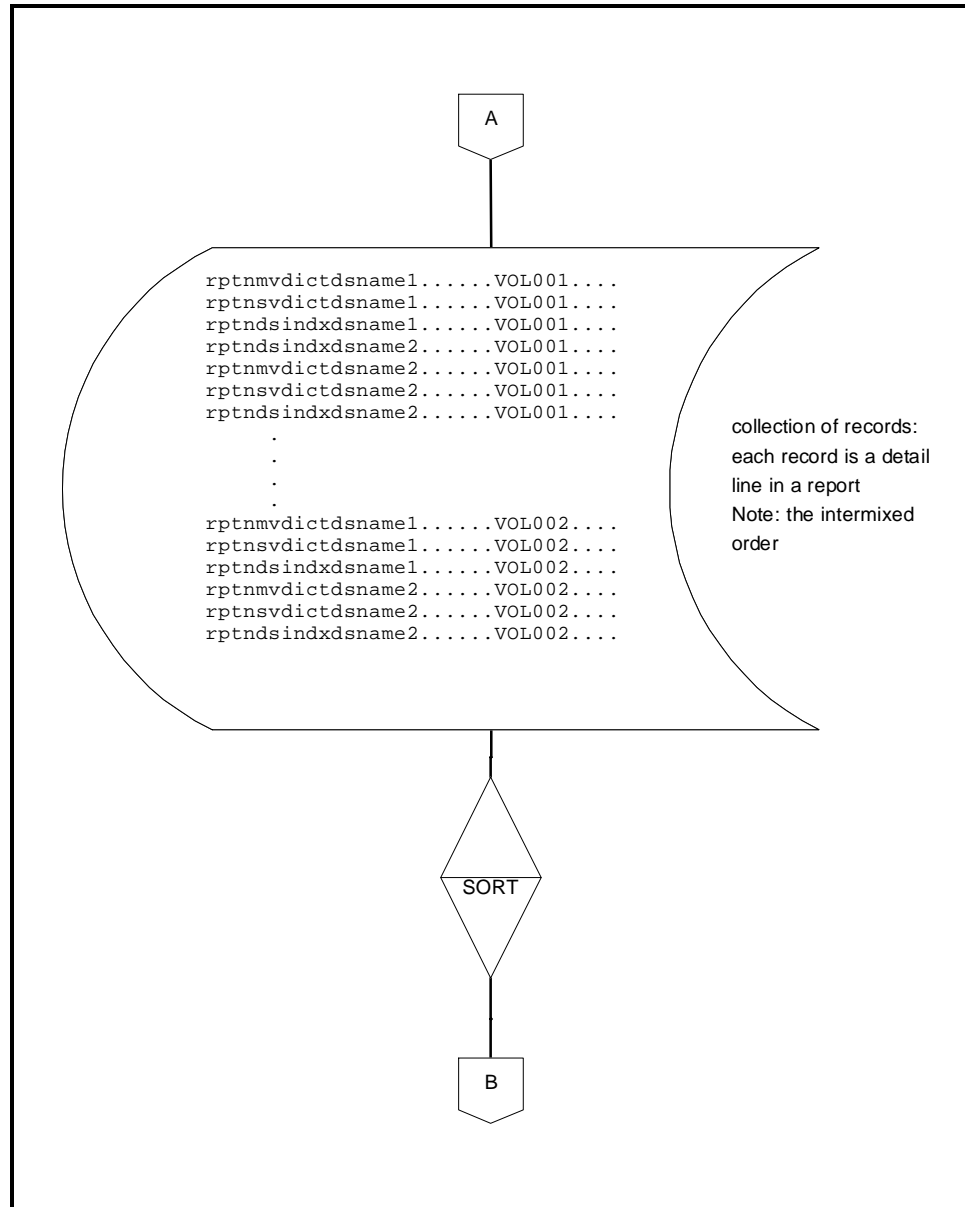


Figure 11-2. Record Collection portion of Report Logic

All detail records for each report are then grouped together as illustrated below:

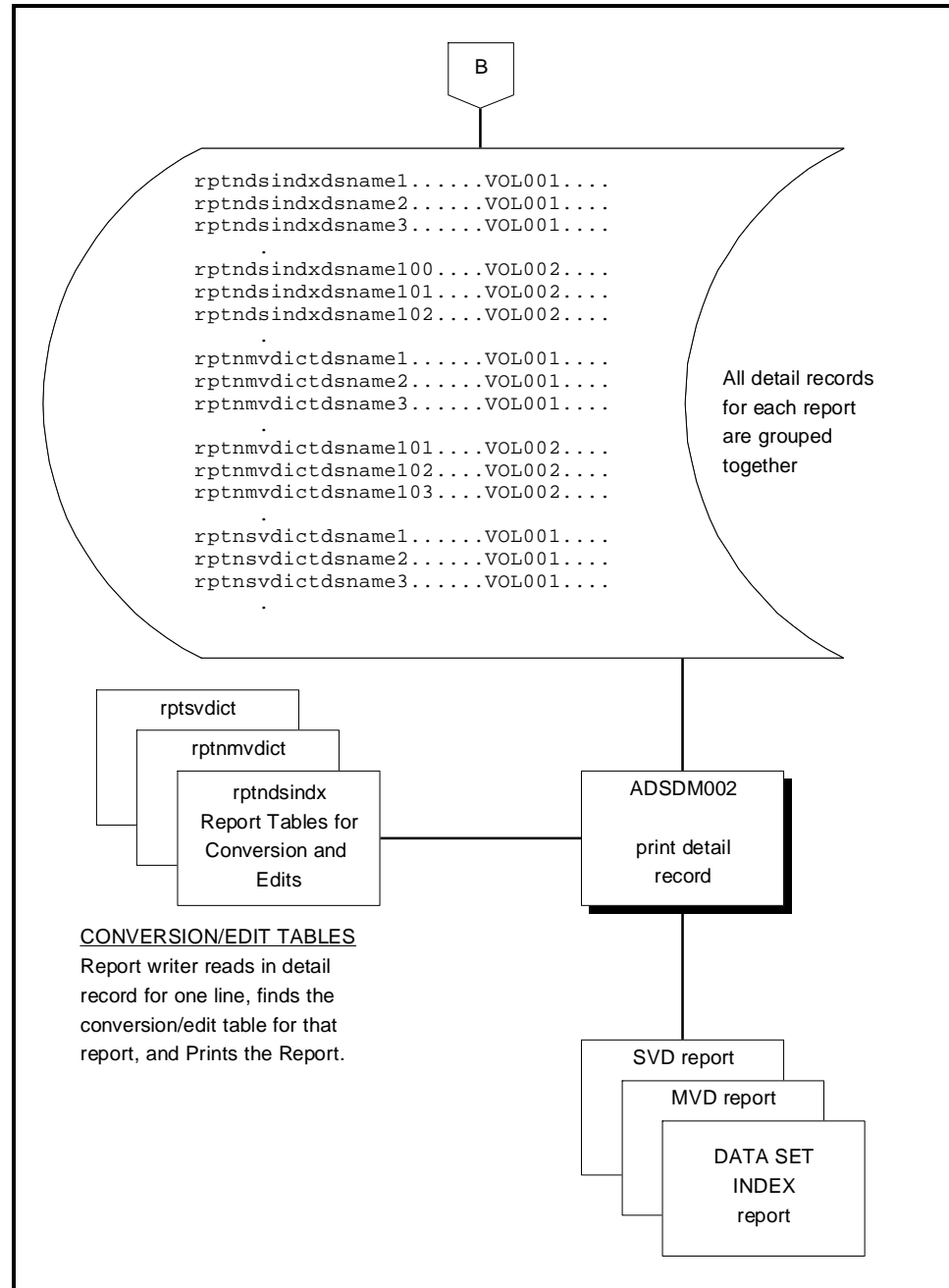


Figure 11-3. Detail Records Grouped Together Example

Report Processing - ISAM

DD Statements and Resulting Allocations

Any ISAM data set greater than one cylinder will always have three separate areas:

```

INDEX AREA ..... I
PRIME AREA ..... P
(Independent) OVERFLOW AREA .. O

```

A one-cylinder data set will never have an I or O area. A fourth area, the Cylinder Overflow (COV), may also exist and appears as the last (xx) number of tracks in each (p) cylinder. Existence of the COV is controlled by the OPTCD and CYLOFL subparameters of the DCB.

When the data set is being created (first allocated), the user may specify explicitly the location and size for each of the three areas. This is done by providing from one to three dd statements, with each dd statement giving the space requirements needed for the area being defined. The PRIME AREA (P) dd statement must always be supplied, with the INDEX (I) and OVERFLOW (O) dd statements needed only if explicit location and/or size is to be controlled. Whatever combinations of the I, P and O dd statements are supplied, they must appear in the order listed.

that is, INDEX DD —before— PRIME DD — before— OVERFLOW DD.

As stated above, the I, P and O areas always exist whether or not they were explicitly allocated via a dd statement. When only a PRIME AREA is allocated, ISAM suballocates cylinders from the end of the PRIME AREA to build the cylinder and master indexes. Tracks for the indexes are taken from the end of the cylinder. The remaining portion of the last cylinder used in this fashion is then made available as an OVERFLOW AREA.

Graphically it appears as follows:

Primary Allocation			
CYL1	CYL2	CYL3	CYL4
			sub-alloc I & O
P	P	P	P
			I

If both index and overflow areas are allocated via dd statements, the indexes and overflow records are placed in those areas, of course. However, if only one of the two is specified, then the other is automatically suballocated as described above;

that is, the indexes are built at the bottom of the last cylinder, and the remainder of the last cylinder is available for overflow.

Primary Allocation			Index	
CYL1	CYL2	CYL3	CYLx	
P	P	P	sub-alloc o	
			O	OVERFLOW suballocated from the Index extent
			I	Index at bottom
COV	COV	COV		

EMBEDDED INDEX: There is one case slightly different from the above rules, which is called the embedded ISAM index. Similar to placing a VTOC in the middle of a DASD pack to reduce head-seeking, the cylinder and master indexes can be placed “in the middle” of the PRIME AREA. The user indicates this request via the SPACE parameter on the dd statement for the PRIME AREA. For example:

```
SPACE=(CYL,(10,,2))
```

would cause the following allocation:

5 CYL PRIME — CYL INDEX — 5 CYL PRIME

If no separate overflow was defined via a dd statement, the unused portion of the **last** Index cylinder would still be designated as OVERFLOW as described above.

ISAM Space Restrictions

Space for ISAM data sets must be allocated on cylinder boundaries and in integral multiples of whole cylinders. The request can be made in either CYL or ABSTR. When ABSTR is used, the cylinder boundaries and multiples requirements must still be met.

Because of the absolute addressing techniques used within ISAM, secondary extents are not allowed. This does not mean, however, that each ISAM area consists of only one extent. The reason for this is found in the DADSM allocation techniques. When a primary space allocation request is made, DADSM tries to satisfy this in a single extent. If a single extent cannot be found to meet the primary request, DADSM will look for two extents that will satisfy the space need. DADSM will continue “dividing” the primary into as many as five extents before giving up and issuing the “space not available” message.

To force DADSM to keep the primary space request in a single extent, the CONTIG parameter must be used in the space request. From this it can be seen that any ISAM data set can have up to five extents on any of its volumes. Since a single ISAM data set can be divided across three volumes (I, P and O), the maximum number of extents associated with the data set would be 15.

Note: This does not include the assumed rare case of a prime area that spans multiple DASD volumes.

VTOC ENTRIES- Format 1, 2 and 3 DSCBs

Allocating a data set is synonymous with creating DSCBs in the VTOC. Every data set has a format-1 DSCB, which will describe up to three extents of the data set. If there are more than three extents (ISAM may have up to five), then the format-3 DSCB will exist. For ISAM, a format-2 DSCB will also exist. It will be found on the volume specified by the first dd statement when the data set was created.

Another way of knowing if a DASD pack should have a format-2 DSCB is to consider the volume sequence (VOLSQ) number. When an ISAM data set is created, the volume specified on the first dd statement is called VOLSQ=1. If the second dd points to a different volume, then it is VOLSQ=2. If the third dd points to yet another volume, then it is VOLSQ=3. In the case where the ISAM areas are spread across two or three volumes, the format-2 DSCB will be found only on the volume with VOLSQ=1.

Note: The format-1 DSCB on the volumes with VOLSQ=2 or 3 will contain the volume serial of the volume with VOLSQ=1; that is, they point back to the volume containing the format-2 DSCB.

Levels of Indexes:

Level	Description
0	Each PRIME cylinder has a track index
1	If there are two or more cylinders of PRIME data, then there is also a cylinder index
2	The user may specify if master indexes are also to be built by use of the (NTM) parameter. When the number of tracks in the cylinder index reaches the (NTM) number, the first level of master index is built.
3	If tracks in this level reach the (NTM), then a second level of master index is built
4	Similarly a third (highest) level can be built

In summary:

LEVELS=0 implies a 1-cylinder ISAM data set
LEVELS=1 implies cylinder indexes exist
LEVELS=2 implies CYL index plus 1st level master
LEVELS=3 implies CYL plus 1st and 2nd level mstr
LEVELS=4 implies CYL and three levels of master

ISAM Report Logic Overview

Space Calculations - INDEX

USED TRACKS:

The format-2 DSCB contains the start and end addresses for each of the cylinder and three levels of master indexes. The sum of the tracks in each range is the number of used index tracks. Special care must be taken if the starting and ending addresses are in two different extents of the data set.

AVAILABLE FOR OVERFLOW:

The fractional part of a cylinder not used by indexes is potentially available as an independent OVERFLOW AREA.

ALLOCATED TRACKS:

The number of tracks actually allocated for use as an INDEX AREA is calculated three different ways, depending on the manner in which it was allocated. The three methods are referred to as I1, I2, and I3 described below:

1. ALLOCATED TRACKS FOR INDEX = SUM OF INDEX EXTENTS

This method is used when a separate index area was allocated and the overflow is not suballocated within this index area.

2. ALLOCATED TRACKS =(SUM OF EXTENTS) - (AVAILABLE OVFLOW)

This method is used when the index area does exist but the overflow is suballocated (does not have its own area).

3. ALLOCATED TRACKS = USED TRACKS

This method is used when the index does not have its own area, but is suballocated from either the prime or overflow areas.

Space Calculations — Prime

Suballocated Tracks (for INDEX and/or OVFLOW): The format-2 DSCB contains a pointer to the last cylinder available for prime data records. If this address is the same as the last cylinder in the prime extents, no suballocation from the prime area has been done. If they are different, the number of cylinders suballocated can be computed, taking special care for the case when the actual end of the available prime area is in a different extent from the end of the allocated prime extents. Graphically,

Primary Allocation (extent 1)			Primary Allocation (extent 2)	
CYL1	CYL2	CYL3	CYL9 suballoc I & O	
P	P	P	O	OVERFLOW suballocated from the Prime extent
COV	COV	COV	I	Index at bottom

In the above diagram, CYL 9 is part of the PRIME AREA, extent two. The actual last cylinder available for prime data is CYL 3 in extent one.

Allocated Tracks:

Allocated Tracks (Prime) = (Sum of Prime Extents) - Suballoc

These calculations are done in cylinders and then multiplied by the number of prime tracks per cylinder, where;

Prime Tracks Per Cylinder = Device Trks/Cyl - (Cov) Trks/Cyl

Used Tracks:

The format-2 DSCB contains the address of the last data record written in the PRIME AREA. This address minus the beginning address of the PRIME AREA EXTENT 1 gives the used area. Again, the calculations must allow for crossing extents, and by doing the calculations in cylinders, the COV areas can be allowed for as described above under “Allocated Tracks”.

Allocated 0 Tracks:

Space calculations for the OVERFLOW AREA can be divided into three cases referred to as O1, O2, and O3 described below:

1. The case when a SEPARATE O AREA exists with no suballocations for index.

ALLOCATED TRACKS = SUM OF THE O EXTENTS

2. The case when a SEPARATE O AREA does not exist, but is suballocated from the INDEX AREA, which may also be suballocated from the PRIME AREA. In this case, the actual available allocated tracks is just the unused portion of the last suballocated cylinder; that is, the portion not used by the indexes. This is the “available overflow” calculated in the Index Space Section.

$$\text{ALLOCATED TRACKS} = \text{AVAILABLE OVERFLOW TRACKS}$$

3. The case when a SEPARATE O AREA does exist, but has a number of tracks suballocated for Indexes.

$$\text{ALLOCATED TRACKS} = (\text{SUM OF THE O EXTENTS}) - (\text{USED I TRKS})$$

Used O Tracks:

The format-2 DSCB contains the address of the LAST OVERFLOW RECORD. For cases O1 and O3, the used tracks are calculated by adding the tracks from the beginning address in EXTENT 1 of the O AREA up to the given address. In case O2, the used tracks is simply the “hh” of the “cchhr” address of the last O record, since that “cc” (of the cchhr) is the only “cc” available as OVERFLOW; that is, the beginning overflow address is cc001 and the last record address is cchhr.

The information to be reported for an ISAM data set can be divided into four major areas:

Table 11-18. Four Major Areas of ISAM data sets

Major Area	Description
Format-1 (and format-3) data	Always available
Format-2 data, direct copy	Available if volume contains a format-2
Detail Space usage data for any of the I, P and O areas	Computable only if the format-2 is on the volume and an analysis of several variables allows it
Other Volume information (Is part of this data set on other volumes?)	An analysis of all the variables will determine what can be reported

ISAM Variables Examined

The VARIABLES ANALYZED to determine how much can be reported are as follows:

1. Does this volume contain extents marked as I, P and/or O?
2. Does this volume contain the format-2 DSCB?

- a. Does the LAST PRIME CYLINDER available equal the last cylinder in the PRIME EXTENTS? (suballocation?)
- b. Does the CYL ADDRESS of the LAST OVERFLOW RECORD equal the cylinder address of the CYLINDER INDEX?

An analysis of all the different possible combinations of the answers to the above will determine the reportable information and hence the logic flow of the ISAM report modules.

Combinations of the Variables

The following chart indicates all of the combinations of VARIABLES on the left side. The right side contains the different information and case number to be reported.

Table 11-19. ISAM Variable Combinations

VARIABLES						REPORTED INFORMATION					
EXTENTS FLAGGED			FMT-2 DSCB	CC P Max=CC P ALLO	CC 0 =CC CI	FMT 1	FMT 2	DETAIL SPACE			Other VOL
I	(P)	O						I	(P)	O	
yes			yes	n/a	no	y	y	I1			other1
yes			yes	n/a	yes	y	y	I2		O2	other2
yes	yes		yes	n/a	no	y	y	I1	P1		other3
yes	yes		yes	n/a	n/a	y	y	I2	P1	O2	other8
yes		yes	yes	n/a	n/a	y	y	I1		O1	other2
yes	yes	yes	yes	n/a	n/a	y	y	I1	P1	O1	other8
	yes			n/a	n/a	y					other4
	yes		yes	yes	n/a	y	y		P1		other5
	yes		yes	no	n/a	y	y	I3	P2	O2	other8
	yes	yes		n/a	n/a	y					other6
	yes	yes	yes	n/a	n/a	y	y	I3	P1	O3	other8
		yes		n/a	n/a	y					other7

In the program logic, the DSCBs for each ISAM data set are examined for the conditions above. The six variables in the table are kept as the six high-order bits in a flag byte that is then used to control logic flow for computing the reportable in-

formation. Each YES position is a 1 bit, each blank or N/A is a 0 bit. For example, the first combination in the table creates a flag byte of x'90', the last combination is x'20'.

Each vertical line in the previous table corresponds to a bit setting in the flag byte. The following 12 values correspond to each line of the table, and are the actual bit setting for that line. (First is top line.)

1. X'90'
2. X'94'
3. X'D0'
4. X'D4'
5. X'B0'
6. X'F0'
7. X'40'
8. X'58'
9. X'50'
10. X'60'
11. X'70'
12. X'20'

ISAM Format-2 DSCB Offsets

The following tables summarize significant fields in the format-2 DSCB used for SPACE CALCULATIONS.

Which Offset	Cyl Indx	1st M	2nd M	3rd M
Starting address	72	79	1	13
Ending address	106	111	8	20

The START addr offset points to an (mbbcchh) field. If the first (b) byte of this address is not equal to zero, the end address given is in a different extent of the data set; that is, the index itself is split across different extents of the data set.

Offset	Overflow (independent)
of Start Addr	---
of Last Record Written Addr	116 (mbbcchhr)
to Free Tracks (RORG2)	126
to Number of O records	128
to Number of O reads/writes (RORG3)	61
	Cylinder Overflow
to Number of Full Areas	130
to Number of TRKS/Area	52
	Prime
to Last TRK of Last CYL avail.	36 (mbbcchhr)
to Last Record Written	93 (mbbcchhr)
to Number of records	67
to Number of DELETE records	59

Chapter 12. Algorithms

Space allocation for a non-VSAM data set is computed in the following manner.

$$\# \text{ of new tracks} = \frac{\text{total bytes of user data in input data set}}{\text{bytes of user data that fits on 1 out track}}$$

A SIZEIN subroutine computes the (*TOTAL BYTES IN INPUT data set*). A SIZEOUT subroutine computes the (*BYTES FITTING ON 1 OUT TRK*) and then performs the division.

Note that the largest result is obtained by maximizing the numerator and minimizing the denominator for this calculation.

After the computation is done, a percentage increase will be made based on sysparm values for ADPCTPDS and ADPCTOTH. The number of tracks as indicated by sysparm ADTRACKS is then added. Finally, if CYL or ROUND allocation is required, the total tracks is rounded up to an even multiple of cylinders.

Subroutine SIZEIN performs its calculations in the following manner:

$$\text{User Data Bytes} = (\text{user data bytes per trk})(\# \text{ of tracks})$$

1. (*# of tracks*) is a specific value passed as input.
2. (*user data bytes per trk*) depends on several items and is estimated as follows, with varying degrees of accuracy.
 - a. (blksiz)(blks per track), — if $0 < \text{BLKSIZ} \leq \text{MAXBLK}$ or
 - b. (maxblksz to fit)(1 per trk), — if not the above or
 - c. track size, — if DSORG=PS/PO, RECFM=V/U, and DEVTYPE changed.

Note: PS data sets will always fit on the same number of tracks if the same device type is used. If a device-type change occurs, variable and undefined-length records may fit unpredictably, so an attempt is made to error on the safest side possible; that is, assume every single byte on the track was a user data byte.

All PO data sets are much like PS RECFM=V/U simply because of multiple members with each member's last block truncated, and the end-of-file marks embedded throughout the data set. Even if the members are copied to the same device type, more space may be required if the order in which the members are physically arranged on the tracks is changed. A device type change further complicates the ability to accurately predict the correct space requirements. For this reason, sysparms ADPCTPDS and ADTRACKS have been provided to automate a dynamic increase in the predicted space quantity.

Bytes that Fit on an Output Track

The number of bytes to fit on an output track is determined as follows:

1. (blksize)(blks per track), — if $0 < \text{BLKSIZE} \leq \text{MAXBLK}$ or
2. (maxblksz to fit)(1 per trk), — if not the above

Special Case:

Number of tracks needed = number of tracks on input, if the DSORG=DA and relative track addressing is used. SAMS:Disk attempts to maintain the TTR position of the data records and hence requires the same number of TT tracks.

VSAM Idle Space Calculation

The idle space that is printed on the VSAM cluster detail report (CLD) and on the multiple and single volume dictionary reports (MVD and SVD), are estimates of free space based on catalog information for the cluster. Its calculation is based on the high used and high allocated relative byte addresses for each of the extents in the data set. It must be kept in mind that this value is only an estimate of idle space, and its accuracy will be dependent on the type of cluster being processed. This value will always express the MINIMUM amount of free space that a cluster has. Even if the idle space displayed is zero, the cluster may indeed have a large amount of embedded free space, especially in a KSDS that was defined with freespace percentages.

To calculate the amount of estimated free space a cluster has, SAMS:Disk retrieves all of the extent information for the data component of the cluster (and also for the index component if the cluster is a KSDS). The high used and high allocated relative byte addresses (RBA's) for each extent are compared to determine the percentage idle space in the extent. A running total is kept for all extents in the cluster. For the MVD and SVD reports, the percentage is applied at the volume level. The CLD report, however, applies the percentage to the data set as a whole, regardless of the number of volumes on which the data set resides. Because of this fundamental difference in reporting algorithms, the idle space total found at the end of the CLD report may vary slightly from that found in the MVD and SVD reports for the same clusters. This variation is insignificant and should be viewed in the light that these values are indeed only estimates. None of the reports can be considered the "most reliable" since the accuracy will vary between clusters.

The accuracy of the estimated idle space value depends on the type of cluster being viewed and the data that comprises the cluster. For an entry-sequence data set (ESDS), the value will be quite accurate. Because there can not be any embedded free space in this type of cluster, the high used and high allocated RBAs will give an accurate depiction of the amount of space being used in the cluster. For a relative-record data set (RRDS), the accuracy will depend on the individual cluster. The idle space will reflect the amount of space available BEYOND the last record

in the data set — it does not account for any empty record slots embedded in the cluster.

The calculation for key-sequenced data sets (KSDS) gives only the amount of idle space beyond the high used relative byte address in each extent. If a cluster is defined with embedded free space, or if the cluster has gone through numerous control interval and control area splits, the amount of actual idle space in the cluster may be significantly more than that calculated. Again, the idle space displayed will be the minimum amount that can be expected in the cluster.

Obviously this idle space calculation cannot be relied upon to give an absolute estimate of unused space in a cluster. Its true value is in pointing out clusters that may, in fact, be overallocated. If the idle space displayed is high, and the cluster is a KSDS or RRDS, then the chances are good that the cluster has too much space allocated to it. These particular clusters should then be examined to determine if they should indeed be reduced in size.

VSAM New Name Algorithm

When a cluster is restored with a new name, SAMS:Disk must determine a new name to use for the data component of the cluster. In addition, if the cluster is a key-sequenced data set, a new name must be generated for the index component. If the data (or index) component name was originally generated by VSAM (that is the NAME parameter was not specified for the data and index components when the cluster was defined), SAMS:Disk will let VSAM generate new name(s) for the cluster's components.

If the data and index components were defined with explicit names when the cluster was created, SAMS:Disk must generate new names for the components to avoid a “duplicate entry” error when defining the new cluster. The base cluster will be named whatever is specified on the NEWNAME= parameter. The data and index component names will be generated based on a comparison of the original cluster name and the newname that was specified. The base portion of the component name will be identical to the newname specified on the input command. The remainder of the name will be generated by comparing the original base cluster name with the original component name. SAMS:Disk will compare the two names until a difference is detected. The portion of the component name that is different is then appended to the base cluster's new name to determine the new name for the component. Note that if the comparison between the original base cluster name and the original component name is at other than a “.”, a “.” will be inserted into the component new name before appending the remaining portion of the component name. This is done to ensure that the name being generated will not exceed eight characters in length.

As an example, let's look at how the following new name would be generated:

```
Original cluster name = LABS.TJP.PMRFILE
Original data component name = LABS.TJP.PMRFILE.DATA
```

If the new name specified is LABS.NEW.PMRFILE, the new name for the data component would be created as follows:

1. The new name is moved into a work area: LABS.NEW.PMRFILE
2. The original cluster name is compared to the original data component name. A difference is detected after the PMRFILE node. The differing characters are: .DATA
3. The differing characters are appended to the base cluster's new name in the work area to determine the component's new name: LABS.NEW.PMRFILE.DATA

Data Compression/Decompression

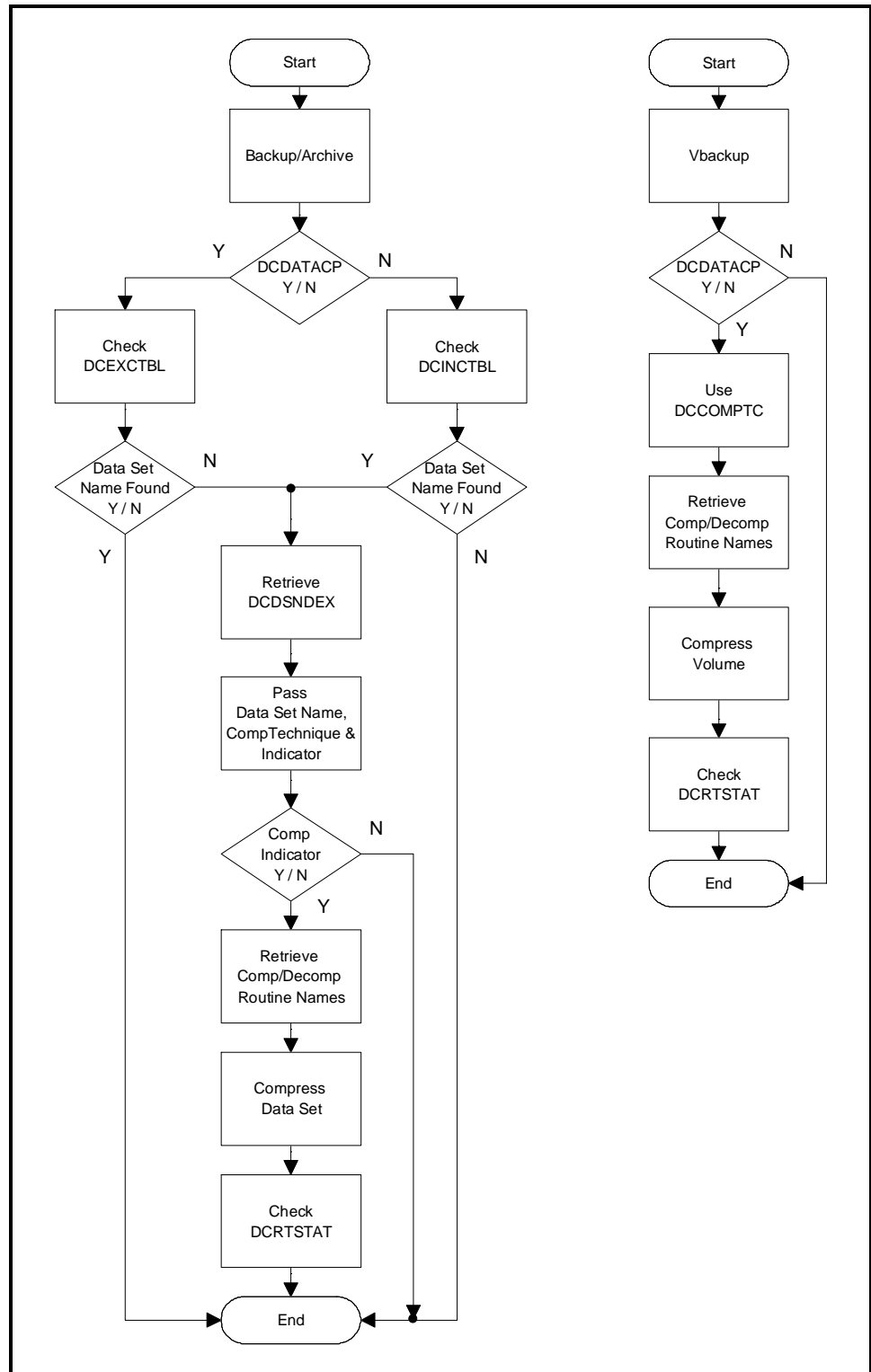


Figure 12-1. General flow of data compression

SAMS:Disk does data compression and decompression at the archive logical record level. Only the data after the eight-byte prefix is compressed (see the archive record format). A high-level control module is called before moving the archive record to the archive block. It decides whether or not to do data compression when it begins processing for each data set.

Sysparm DCDATACP is used to set the initial value of the compression indicator. If this is not a volume-level backup, the exclusion/inclusion table is referenced. If DCDATACP is specified with a value of Y but the data set is in the exclusion table, the compression indicator will be set to N (no). If sysparm DCDATACP is specified with a value of N but the data set is in the inclusion table, the compression indicator will be set to Y (yes). Inclusion and exclusion tables can not be used at the same time.

The compression technique number is indicated by sysparm DCCOMPTC. The sysparm default is always selected for volume-level backup.

If user exit DCDSNDEX is active, the technique number, yes/no compression indicator, and the data set name are passed for the exit to make the decision. Each technique number corresponds to two exits—one for compression (DCCMPEXn) and one for decompression (DCDCPEXn). Both must be present at compression time.

If compression statistics are desired, sysparm DCRTSTAT can be specified with a value of Y for archive volume totals or D for data set totals and archive volume totals.

Each compressed record is returned with a two-byte prefix. The first character is X'EE'. It is for doublechecking and indicates the record is in compressed format. The second character is X'Fn', n being any character A-F or 0-9, to indicate the compression routine used. This character allows multiple compression routines to be implemented for the same application. This is useful when you have knowledge of the data and know that specific routines handle specific kinds of data more efficiently. Once data has been compressed using a given compression technique, that technique must remain the same.

Data Compression Techniques

There are 16 data compression/decompression exits to chose from. Technique exits 0-7 are reserved for SAMS:Disk, while technique exits 8-F are available to the user. The following table is a list of the 16 exits:

Table 12-1. Data Compression Techniques available

Technique	Compression Exit	Decompression Exit
0	DCCMPEX0-ADSUT371	DCDCPEX0-ADSUT372
1	DCCMPEX1-ADSUT410	DCDCPEX1-ADSUT411
2	DCCMPEX2	DCDCPEX2
3	DCCMPEX3	DCDCPEX3
4	DCCMPEX4	DCDCPEX4
5	DCCMPEX5	DCDCPEX5
6	DCCMPEX6	DCDCPEX6
7	DCCMPEX7	DCDCPEX7
8	DCCMPEX8	DCDCPEX8
9	DCCMPEX9	DCDCPEX9
A	DCCMPEXA	DCDCPEXA
B	DCCMPEXB	DCDCPEXB
C	DCCMPEXC	DCDCPEXC
D	DCCMPEXD	DCDCPEXD
E	DCCMPEXE	DCDCPEXE
F	DCCMPEXF	DCDCPEXF

Data Compression Technique 0 (DCCMPEX0)

This compression routine compresses replicated spaces, binary zeros and other replicated character strings. It starts with a 126-byte fixed-length character string (or the length of the data, whichever is shorter).

Typical flow of this process is illustrated in the figure below:

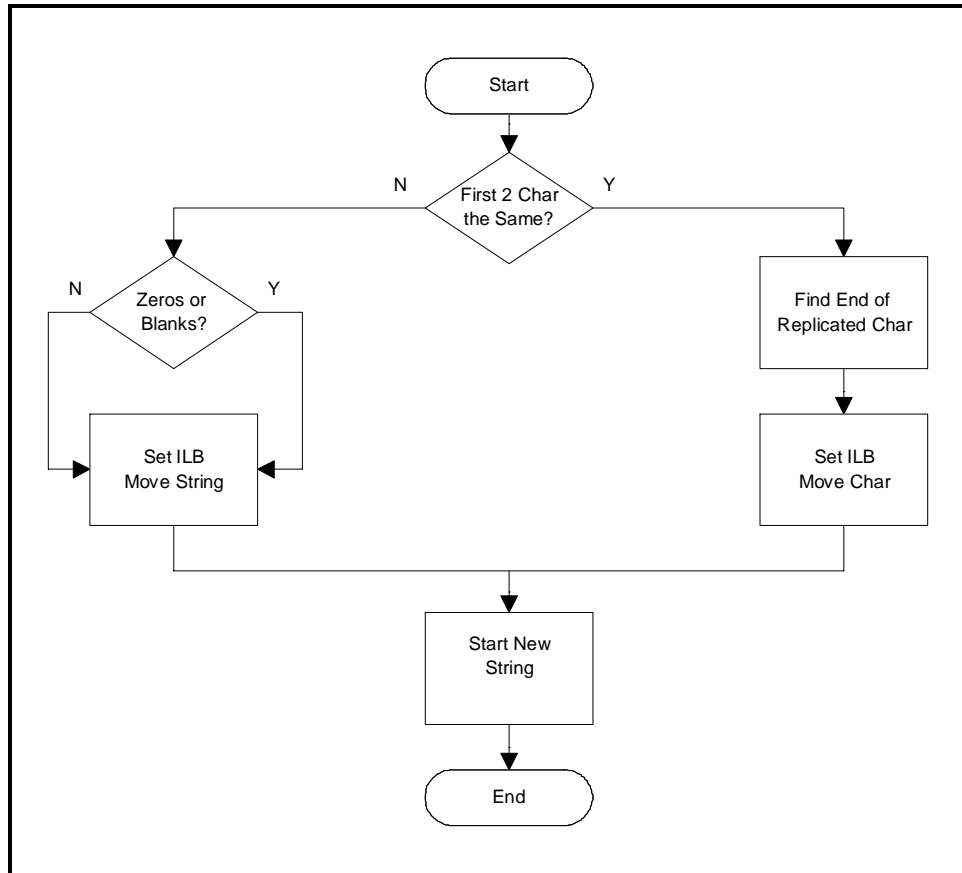


Figure 12-2. Data Compression Technique 0 Flowchart

INDICATOR/LENGTH BYTE

ILLLLLLL

I - String indicator bit

1 - Replicated string

0 - Non-replicated string

LLLLLLL - LENGTH BITS

The max string length is 126 characters. If there are more than 126 characters in a string, a new character string is started.

Due to the storage technique used, it is possible to compress almost all replicated character strings to 2 characters. For performance reasons, it is not practical to test for all possible replicated character strings. This routine specifically looks only for replicated spaces and binary zeros.

Example 1:

The following string will be compressed as shown. This example shows compressing 40 bytes to 30.

```

                MVC      RESULT,=H'1'          SET CC
-----1-----2-----3-----4

|  |  |  | MVC |  |  | RESULT,=H'1' |  |  | SET CC |
| 8B|40|03|____|83|40|12|_____|87|40|06|_____|
BYTE 1  2  3 4-6  7  8  9      10-21      22 23 24  25-30

```

Example 2:

The following string will be compressed as shown. This example shows compressing 130 bytes to 7. The 126 spaces are compressed to a replicated string. The space and the 3 asterisks become a 4-byte literal.

```

                                     ***
-----1-----2-----1-----1
                                     2       3
                                     0       0

|  |  |  | *** |
| FE|40|04|____|
BYTE 1  2  3  4-7

```

Example 3:

The following string will be compressed as shown. this example shows compressing 130 bytes to 4. The 126 spaces are compressed to a replicated string. The 4 asterisks are compressed to a replicated string.

```

                                     ****
-----1-----2-----1-----1
                                     2       3
                                     0       0

|  |  |  | * |
| FE|40|84|__|
BYTE 1  2  3  4

```

Compression Results

Our TSS type packs averaged 55 percent savings in space. The CPU time almost doubled, but the elapsed time remained about the same. Our system packs averaged 30 percent savings in space. The CPU time increased by a fifth and the elapsed time increased by a third.

Overview of Space Allocation for Archival to DISK

SAMS:Disk determines the amount of space to allocate (the amount of space it needs on a disk pool volume) in the following manner. Quite frequently the primary output will be on disk and the duplicate will be on tape or a F6470 cartridge. The output archive type dictates the block size to be used. This, coupled with the density and length of tape being used, allows an easy and very accurate calculation of the tape capacity. (Since the ends of F6470 tape cannot be clipped off, the most common variable in the calculation has been removed!)

The optimal disk allocation is exactly the same capacity as its backup medium, such that when one is filled and both must be closed, nothing is wasted on the other medium. To do this on an exact equivalence, however, would require that very large amounts of free space be available. Therefore SAMS:Disk selects 1/16th of that value as the default primary and secondary space allocations. In many cases, an archive run will not need a full tape, and one or two extents will be more than enough space. However, when a large archive run is made and the disk pool volumes contain plenty of free space, SAMS:Disk will obtain as many as possible of the 16 extents before closing the archive data set. This maintains high tape usage as well. Any excess (unused) disk space is, of course, released immediately when the data set is closed.

SAMS:Disk also considers the size of the data sets being archived when determining the amount of space to allocate. If the first data set to be archived is larger than the default primary space, the primary space is reset to the size of the data set. If free space equal to this new primary value is not available but the default value is, as a last resort SAMS:Disk will attempt to archive the input data set to a multivolume output data set.

Once the target archive data set is allocated and the first data set to be archived is copied to it, SAMS:Disk will continue to copy additional data sets (the second through nth) as long as there is sufficient space to hold them. To determine whether sufficient space is available, SAMS:Disk examines the unused space in the current extent, and checks to see if additional extents can be obtained. If there is not sufficient space, the current disk archive data set is closed and a new one is allocated, much like swapping to a new tape.

Space Allocation Algorithm

To begin the creation of a disk archive data set SAMS:Disk issues an enqueue against the SAMS:Disk DASD POOL to make sure that other SAMS:Disk jobs wait until allocation to a volume within the pool is complete. SAMS:Disk then saves the input data set size, and determines the minimum primary and secondary allocation requirements specified by sysparms SPACEPRI and SPACESEC. When these sysparms are set to their default values, minimum allocation requirements are determined by tape capacity in bytes divided by 16. Tape capacity is calculated based on device type and density to get bytes per foot and multiplied by the number of feet indicated by sysparms TAPEFEET or CARTFEET. The default for this cal-

ulation is TAPEFEET unless F6470s are indicated on the ARCHIVEC dd card. If SPACEPRI and SPACESEC are not set at their default values, the minimum allocation requirement is the value * 1024 * 1024. If packing is not used (sysparm ARCHPACKn) the input data set size becomes the minimum allocation requirement. Otherwise the input data set size is used only if it is greater than the calculated minimum value.

If the primary and copy cannot reside on the same volumes as indicated by sysparm ARCSEPC0 then the primary volume list is added to an exclude volume list used during volume selection to insure they are not selected for the copy.

SAMS:Disk now begins to build a list of volumes, the volume with the largest amount of free space being first, the next largest being second and continuing until the number of bytes in the data set has been satisfied or until the maximum number of volumes, or the number of volumes in the pool if less than 35, has been reached.

The allocation satisfied the first time through is the primary amount, if successful, the volume is added to the list and to the exclusion list, and the allocation to satisfy becomes the secondary allocation. If the primary amount cannot be satisfied with the first volume returned, and the free space associated with that volume is less than the minimum allocation amount for primary, an abend will occur. If the available free space is greater than the calculated minimum allocation, the data set size is divided by 35, or the number of volumes in the pool if less than 35, and becomes the new minimum primary amount. If the returned free space is still less than the new minimum primary amount, an abend will occur. If not, the primary amount will be set to the returned free space, and will be subtracted from the data set size. The remaining bytes will be divided by 35 (or the number of volumes in the pool if less than 35) minus one to become the new minimum secondary amount. If the new minimum secondary amount is greater than the calculated secondary allocation, the value to satisfy is the minimum secondary allocation. The volser is added to the volume and exclusion lists and processing continues.

The secondary allocations begin when the primary has been satisfied. Only when a volume's free space cannot satisfy the secondary allocation amount or if the input data set size has been satisfied will the volume list be complete. If there are no volumes in the list at this time, an abend will occur. If there is a list of volumes, they are returned to the calling program so that allocation can take place. When allocation is complete a dequeue is issue against the SAMS:Disk DASD POOL.

General Comparison of Allocation in Prior Releases

1. The primary allocation value is still set to 1/16 of tape/cart, or the value of SPACEPRI if specified. (No change here.)
2. Release 7.6 and prior SAMS:Disk takes the first volume from the pool that satisfies SPACEPRI for contiguous free space. If more than one SAMS:Disk job looks at the same pool of volumes, each may examine volume space at the same time, unaware that the other sees the identical "space status" for the volume. The first job to allocate should be

successful, but the others risk even a primary allocation error. If all concurrent primaries succeed, the probability of getting a secondary allocation is certainly decreased for all of the jobs!

Release 7.7 or higher begins by serializing the examination of the archive pool and the actual allocation of the space. This is done by an ENQ on Major=DMSOS, Minor=ARCHPOOL when the pool is examined, with the DEQ done after the allocation has taken place. Release 7.7 or higher also examines all volumes in the pool and selects the volume with the greatest contiguous free space. This ensures the primary allocation and increases the probability of getting extra extents if needed.

By serializing the selection and allocation among concurrent SAMS:Disk jobs, each job will see a different "space status" for the pool volumes, increasing the distribution across different volumes. If different primary volumes are selected (with increased probability), contention for secondary extents is reduced.

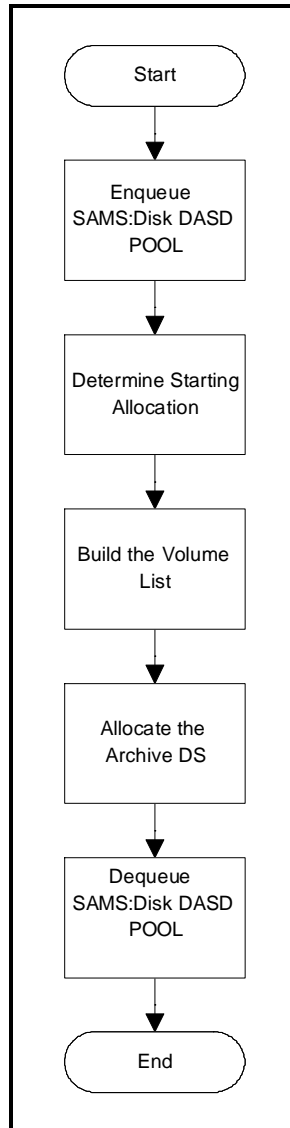
3. If the size of the input data set is larger than the size in item 1 above (1/16 tape or SPACEPRI), the data set size is compared to the largest contiguous free space that was found. If the free space is big enough, the primary is set to the data set size and allocated. This is the same as in prior releases, except that the probability is greater that the data set will fit, because the volume with the largest contiguous free space was selected first.

If the data set size is greater than the largest contiguous free space, additional volumes are selected with the 2nd, 3rd, and so on largest contiguous free space until the size of the data set can be satisfied, and a multivolume allocation is done. If the pool is exhausted without finding enough contiguous space to satisfy the data set, the archive is still attempted on the theory that additional extents from non contiguous free space on each of the selected volumes may be able to satisfy the request. If they do not, a B37 abend for the data set being processed will occur. It will be bypassed as "too large" for the archive pool, and processing continues with the next data set. Again, the significant difference in 7.7 or higher is that each volume selected from the pool is the one with the largest contiguous free space, the next largest, and so on. In 7.6, it was a sequential search of the pool, taking any volume that met the minimum secondary space need. Now we get the volume with the most space that meets that need.

Creating a Disk Archive Data Set

1. SAMS:Disk issues an enqueue against the SAMS:Disk DASD POOL, major name = DMSOS, minor name = ARCHPOOL.
2. Determine the primary and secondary values for the archive data set. If sysparms SPACEPRI and SPACESEC are defaulting they are recalculated to tape capacity in bytes divided by 16. If a value is specified, it is multiplied by 1024 and again by 1024 (SPACEPRI*1024*1024).
3. Determine the starting primary allocation. Primary will be the above calculated SPACEPRI value unless data sets are not packed and the data set size is less than the calculated SPACEPRI value, in which case primary will be set to the data set size. If SPACEPRI is still smaller than the data set size, primary will be set to the data set size.
4. Begin an exclusion list of volsers to ensure integrity as selections are being made. If this archive data set is for the copy, and sysparm ARCSEPC0 is set to yes, then the list of primary volsers are added to the exclusion list to insure they are not re-selected to be the copies.
5. Set the desired number of bytes to the calculated primary value.
6. Build the volume list. Continue building until the limit of 35 (or the number of volumes in the pool if less than 35) has been reached, or the number of bytes in the initial data set has been satisfied in either single or multiple volumes. The building of the volume list abends under the following conditions:
 - a. If the returned volume does not satisfy the desired number of bytes, and it is the first volume, and returned free space does not satisfy SPACEPRI needs.
 - b. If the returned volume does not satisfy the desired number of bytes, and it is the first volume, and the data set size divided by 35, or the number of volumes in the pool if less than 35, does not satisfy the minimum primary allocation specification.
 - c. If a volume was not found that was not in the exclude list or of the correct device type or if it's largest contiguous space could not satisfy the desired number of bytes requested.
7. Return the list of volumes.
8. Allocate the archive data set.
9. Issue a dequeue against the SAMS:Disk DASD POOL.

This process is illustrated below:



PDS Compression Threshold Calculations

Calculating compression thresholds for a PDS is divided up into two categories:

1. Directory Used Space Percentage
2. Member Data Used Space Percentage

Directory Used Space Percentage

$$PCT\ Used = \frac{((DBUSE-1)*256+NOBDB-14)*100}{(DBUSE+DMBTY)*256}$$

or

$$PCT = \frac{\begin{array}{l} \text{(total bytes used in all full directory block)} \\ + \text{(total bytes used in last used directory block)} \\ - \text{(14 bytes for the directory end-of-file)} \end{array}}{\text{total bytes allocated}}$$

Where:

PCT USED — percentage of used directory block space calculated in bytes

DBUSE — number of used directory blocks

NOBDB — number of bytes used in the last used directory block

DMBTY — number of empty directory blocks

Member Data Used Space Percentage

$$PCT\ Used = \frac{((DSNTRK-DIRTRK)*TRKLN-DSNBAL+DIRBAL)*100}{(EXTTRK-DIRTRK)*TRKLN+DIRBAL}$$

or

$$PCT = \frac{\begin{array}{l} \text{(total bytes used for all full data tracks)} \\ - \text{(unused bytes on last used data track)} \\ + \text{(used member data bytes on track sharing the directory)} \end{array}}{\text{total bytes for all tracks allocated}}$$

Where:

PCT USED — percentage of member data used space calculated in bytes

DSNTRK — relative track of the last data record

DIRTRK — relative track of the directory end-of-file

TRKLN — track capacity of the device

DSNBAL — track balance of the last data record or DS1TRBAL

DIRBAL — track balance of the directory end-of-file

EXTTRK — relative track of the last allocated track

PDS Compression Space Calculations

Calculating compression space for a PDS is divided up into three categories:

1. Directory Block Allocation
2. Member Data Space Allocation
3. Secondary Space Allocation

Directory Block Allocation

$$DIRBLKS = \frac{(((DBUSE-1)*256+NOBDB-14)*100)+(14*DD)}{DD*256}$$

or

$$DIRBLKS = \frac{\begin{array}{l} \text{(total bytes used in all full directory blocks)} \\ + \text{(total bytes used on last used directory block)} \\ - \text{(14 bytes for the directory end-of-file)} \\ * 100 \\ + \text{(directory end-of-file length)} \\ * \text{(percentage requested)} \end{array}}{\text{percentagerequested} * \text{directory block size}}$$

Where:

DIRBLKS — number of directory blocks to be allocated

DBUSE — number of used directory blocks

NOBDB — number of bytes used in the last used directory block

DD — requested percentage of directory space to be used after reallocation

Member Data Space Allocation

$$EXTTRK = \frac{((DS1TRK - DIRTRK) * TRKLN + DIRBAL - DS1BAL) * 100}{PP} - DIRBAL + DIRTRK$$

or

$$DIRBLKS = \left[\frac{\left\{ \begin{array}{l} \text{(total bytes used in all full data tracks)} \\ + \text{(total bytes used on partial-data track)} \end{array} \right\} * 100}{\text{(percentage requested)}} - \frac{\text{bytes remaining after the last directory track}}{\text{track capacity}} \right] + DIRTRK$$

Where:

EXTTRK — total number of tracks to allocate for primary space, including space for the directory

TRKLN — track capacity of the device

DS1TRK — relative track of the last record written for the compressed PDS.

DIRTRK — calculated relative track for the new directory's end-of-file

DS1BAL — track balance of the last record written for the compressed directory

PDS DIRBAL — calculated track balance of the new directory's end-of-file

PP — requested percentage of member space to be used after reallocation

Secondary Space Allocation

$$SECTRKS = \frac{((MBRTRK * TRKLN) + (TRKLN - MBRBAL))}{SS * TRKLN}$$

or

$$SECTRKS = \frac{\begin{array}{l} \text{(total bytes used in all full data tracks)} \\ + \text{(total bytes used on the last used data track)} \\ * 100 \end{array}}{\begin{array}{l} \text{percentagerequested * track capacity} \\ ((MBRTRK * TRKLN) + (TRKLN - MBRBAL)) * 100 \end{array}}$$

$$SECTRKS = \frac{\text{SS} * \text{TRKLN}}{\text{SS} * \text{TRKLN}}$$

or

$$SECTRKS = \frac{\begin{array}{l} \text{total bytes used in all full data tracks} \\ + \text{total bytes used on the last used data track} \\ * 100 \end{array}}{\text{percentage requested} * \text{track capacity}}$$

Where:

SECTRKS — number of tracks to request for the secondary allocation amount

TRKLN — track capacity of the device

MBRTRK — relative track of the last record written for the compressed PDS.

MBRBAL — track balance of the last record written for the compressed PDS.

PDS SS — requested percentage of secondary space as a function of the member space used after compression

Notes:

- Percentages are rounded up to the nearest whole point.
- Percentages are based on data size prior to compression.
- One directory block = 256 bytes.
- Allocations are rounded up to the nearest track.

Chapter 13. Record Formats

Report Record Format - (RPTNALLOCS)

Volume Allocation Summary Report — ALLOCS

```
ALLOCS
*
*      Mapping of Report record used to print VOLUME
*      ALLOCATION SUMMARY report
*
DETREC   DS      0F
DETCONLN DS      H      RECORD LENGTH, 74 BYTES
          DS      H      (FILLER, ZEROS)
DETTABLE DS     CL10     REPORT NAME 'RPTNALLOCS'
DETTYPE  DS     XL2      DEVICE TYPE
DETSERAL DS     CL6      VOLUME SERIAL
DETDSCB0 DS      H      DSCB 0 COUNT
DETDSCB1 DS      H      DSCB 1 COUNT
DETDSCB2 DS      H      DSCB 2 COUNT
DETDSCB3 DS      H      DSCB 3 COUNT
DETDSCB4 DS      H      DSCB 4 COUNT
DETDSCB5 DS      H      DSCB 5 COUNT
DETDSCB6 DS      H      DSCB 6 COUNT
DETDEVTK DS     XL4      TRACKS PER DEVICE
DETTKALC DS     XL4      TRACKS ALLOCATED
DETTKAVL DS     XL4      TRACKS AVAILABLE
DETFREXT DS     XL2      EXTENTS
DETTKIDL DS     XL4      TRACKS IDLE
DETFRSPL DS     XL4      1ST LARGEST FREE SPACE
DETFRSPS DS     XL4      2ND LARGEST FREE SPACE
DETFRSPT DS     XL4      3RD LARGEST FREE SPACE
DETVOLAD DS     CL4      VOLUME ADDRESS
DETSMS   DS     CL4      SMS VOLUME STATUS
DETRECLN EQU    * DETREC LENGTH OF RECORD, 74 BYTES
*
```

Table Used for Printing ALLOCS Report

```

MEMBER = RPTNALLO
'C1 058052001132050'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K049 041 V O L U M E A L L O C A T I O N S U M M A R Y'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K015 009 DEVICE UNIT SMS'
'H4 K046 027 ----DATA SET CONTROL BLOCKS (DSCB) COUNTS-----'
'H4 K027 075 -----TRACKS-----'
'H4 K030 104 -----FREE SPACE-----'
'H5 K006 001 VOLUME'
'H5 K015 010 TYPE ADDR STAT'
'H5 K046 027 FREE F1 F2 F3 F4 F5 F6'
'H5 K027 075 DEVICE ALLOC AVAIL IDLE'
'H5 K027 104 EXTENTS THREE LARGEST'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 -----'
'H6 K012 121 -----'
'D1 R006001009 DEV TYPE'
'D1 C006003001 SER'
'D1 C004053016 ADDR'
'D1 K001 020 '
'D1 C004057021 SMS STAT'
'D1 H002009026005Z 0'
'D1 H002011033005Z 1'
'D1 H002013040005Z 2'
'D1 H002015047005Z 3'
'D1 H002017054005Z 4'
'D1 H002019061005Z 5'
'D1 H002021068005Z 6'
'D1 F004023075006Z TRKS DEV'
'D1 F004027082006Z TRKS ALLOC'
'D1 F004031089006Z TRKS AVAIL'
'D1 F004037096006Z TRKS IDLE'
'D1 H002035106005Z EXTENTS'
'D1 F004041113006Z L'
'D1 F004045120006Z S'
'D1 F004049127006Z T'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K010 001 TOTAL VOLS'
'T2 A000000012004Z VOL CNT'
'T2 A000009025006Z 0 DSCB CNT'

```

```

'T2 A000011032006Z  1 DSCB CNT'
'T2 A000013039006Z  2 DSCB CNT'
'T2 A000015046006Z  3 DSCB CNT'
'T2 A000017053006Z  4 DSCB CNT'
'T2 A000019060006Z  5 DSCB CNT'
'T2 A000021067006Z  6 DSCB CNT'
'T2 A000035104007Z  TOTAL NUMBER OF EXTENTS'
'T3 K040    001  -----'
'T3 K040    041  -----'
'T3 K040    081  -----'
'T3 K012    121  -----'
'T4 K023    040  TOTAL TRACKS PER DEVICE'
'T4 A004023072009Z  TOTAL TRKS'
'T5 K040    001  -----'
'T5 K040    041  -----'
'T5 K040    081  -----'
'T5 K012    121  -----'
'T6 K022    046  TOTAL TRACKS ALLOCATED'
'T6 A004027079009Z  TOTAL TRKS ALLOC'
'T7 K009    059  AVAILABLE'
'T7 A004031086009Z  TOTAL TRKS AVAIL'
'T8 K004    059  IDLE'
'T8 A004037093009Z  TOTAL TRKS IDLE'

```

Report Record Format — RPTNAMVDIC

A-FORM of MVDICT Report

```
AMVDIC
*
*      Mapping of report record used to print A-FORM of MULTIPLE
*      VOLUME DICTIONARY report with owner
*      Note: The RPTDSORG field has
*            1. X'02' for PDS and X'FE' for PDSE
*            2. X'40' for PS and X'FC' for PSE
*
AMVDREC DS 0F
RPTRDW DS 2H
RPTTABLE DC CL10'RPTNAMVDIC'
RPTAINFO DS 0CL86
RPTDSNAM DS CL44 DATA SET NAME
RPTVOLNO DS CL6 VOLUME SERIAL
RPTDEVCL DS XL2 DEVICE CLASS
RPTDSORG DS XL2 DSORG
RPTRECFM DS XL1 RECFM
RPTBLKSI DS XL2 BLOCKSIZE
RPTLRECL DS XL2 LRECL
RPTSCALO DS XL1 SECONDARY ALLOCATION
RPTSCQTY DS XL3 AMOUNT OF SECONDARY ALLOCATION
RPTTKSUM DS XL3 TOTAL TRACKS ALLOCATED
RPTCREDT DS XL3 CREATION DATE
RPTEXPDT DS XL3 EXPIRATION DATE
RPTDSIND DS XL1 SECURITY
RPTPRMAL DS XL2 PRIMARY ALLOCATION
RPTIDLET DS XL2 IDLE TRACKS
RPTVOLAD DS CL4 VOLUME ADDRESS
RPTCATLG DS CL1 CATALOG STATUS
RPTLUSDT DS XL3 LAST USED DATE
RPTNOEXT DS XL1 NUMBER OF EXTENTS
RPTAOWNR DS CL11 OWNER
```


Table Used to Print MVD Report by Owner

```

MEMBER = RPTNAMVD
'C1 058052001132101'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 130'
'H1 K033 032 M U L T I P L E V O L U M E D'
'H1 K037 065 A T A S E T D I C T I O N A R Y'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K006 046 DEVICE'
'H4 K029 081 -ALLOCATION - TRACKS-'
'H5 K013 001 DATA SET NAME'
'H5 K005 030 OWNER'
'H5 K006 039 VOLUME'
'H5 K016 046 TYPE CAT DSORG'
'H5 K029 063 RECFM BLKSZ LRECL TYPE PRIM'
'H5 K016 094 SEC ALLOC IDLE'
'H5 K021 112 EXT CREDIT EXPDT USED'
'H6 K040 001 _____'
'H6 K040 041 _____'
'H6 K040 081 _____'
'H6 K012 121 _____'
'D1 C044001001025 DSN'
'D1 C011087027 OWNER'
'D1 C006045039 VOL'
'D1 R002051046 DEVICE TYPE'
'D1 C001082054 CAT'
'D1 V002053058 DSORG'
'D1 W001055064 RECFM'
'D1 H002056069005Z BLKSIZE'
'D1 H002058075005Z LRECL'
'D1 U001060082003 ALLO TYPE'
'D1 H002074086005Z ALLO PRI'
'D1 B003061092005Z ALLO SEC'
'D1 B003064099005Z ALLO SUM'
'D1 H002076105005Z ALLO IDL'
'D1 B001086112003Z EXT NO'
'D1 Y002067116 CREATE'
'D1 Y002070122 Z EXPIRE'
'D1 Y003083128 Z LAST USED'

```

```
'T1 K040 001 _____'  
'T1 K040 041 _____'  
'T1 K040 081 _____'  
'T1 K012 121 _____'  
'T2 K015 010 TOTAL DATA SETS'  
'T2 A000000026007Z COUNTER'  
'T2 K022 072 TOTAL ALLOCATED TRACKS'  
'T2 A000065095009Z ACCUMULATOR'  
'T3 K017 072 TOTAL IDLE TRACKS'  
'T3 A000076101009Z ACCUMULATOR'
```

Report Record Format — (RPTNATTRBS)

Volume Attribute Summary Report

```

ATTRBS
*
*      Mapping of report record used to print VOLUME ATTRIBUTE
*      report
*
DETREC    DS      0F
DETCONLN  DS      H      RECORD LENGTH, 60 BYTES
          DS      H      (FILLER, ZEROS)
DETTABLE  DS      CL10    REPORT NAME, 'RPTNATTRBS'
DETDEVCL  DS      XL2     DEVICE CLASS
DETVOLNO  DS      CL6     VOLUME SERIAL
DETADDRS  DS      CL3     UNIT ADDRESS
DETSUBMD  DS      C      SUB MODULE
DETDEVST  DS      XL1     DEVICE STATUS
DETVOLST  DS      XL1     VOLUME STATUS
DETDISOIS DS      H      ISAM DATA SET COUNT
DETDOPS   DS      H      PS DATA SET COUNT
DETDODA   DS      H      DA DATA SET COUNT
DETDOPPO  DS      H      PO DATA SET COUNT
DETDSOAM  DS      H      VSAM DATA SET COUNT
DETDROOT  DS      H      UNKNOWN DATA SET COUNT
DETDSTOT  DS      H      TOTAL DATA SETS
DETDSCBV  DS      H      DSCB COUNT
DETDSCBA  DS      H      ALLOCATED DSCB COUNT
DETDEVTK  DS      XL4     TOTAL TRACKS FOR DEVICE
DETALOTK  DS      XL4     TOTAL TRACKS ALLOCATES
DETALPCT  DS      H      PERCENT ALLOCATED
DETSMS    DS      CL4     SMS VOLUME STATUS
DETRECLN  EQU    * DETREC LENGTH OF RECORD, 60 BYTES
*

```

Table Used to Print ATTRIBS Report

```

MEMBER = RPTNATTR
'C1 058052001132'
'H113132 001'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K047 043 VOLUME ATTRIBUTE SUMMARY'
'H2 4132 001'
'H3 K001 001132'
'H4 K039 009 DEVICE UNIT SMS MOUNT ALLOCATION'
'H4 K039 050 -----DATA SET ORGANIZATION-----'
'H4 K009 092 --TOTAL--'
'H4 K013 104 ----DSCBS----'
'H4 K013 120 ---TRACKS---'
'H5 K006 001 VOLUME'
'H5 K004 010 TYPE'
'H5 K031 016 ADDR STAT STATUS ATTRIBUTE'
'H5 K037 052 PS DA PO IS AM UN'
'H5 K009 092 DATA SETS'
'H5 K013 104 VTOC ALLOC'
'H5 K013 120 DEVICE ALLOC'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 -----'
'H6 K012 121 -----'
'D1 C006003001 SER'
'D1 R002001009 DEVICE TYPE'
'D1 C004009016 ADDRESS'
'D1 K001 020 '
'D1 C004043022 SMS STATUS'
'D1 1001013028 MOUNT STAT'
'D1 2001014039 ALLOC ATTRIB'
'D1 H002017049005Z PS'
'D1 H002019056005Z DA'
'D1 H002021063005Z PO'
'D1 H002015070005Z IS'
'D1 H002023077005Z AM'
'D1 H002025084005Z UN'
'D1 H002027094005Z DATA SETS'
'D1 H002029104006Z VTOC'
'D1 H002031111006Z ALLOC'
'D1 F004033120006Z DEVICE'
'D1 F004037127006Z ALLOC'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K010 001 TOTAL VOLS'
'T2 A000000012004Z VOL CNT'

```

```

'T2 K006    035    TOTALS'
'T2 A000017048006Z  PS CNT'
'T2 A000019055006Z  DA CNT'
'T2 A000021062006Z  PO CNT'
'T2 A000015069006Z  IS CNT'
'T2 A000023076006Z  AM CNT'
'T2 A000025083006Z  UN CNT'
'T2 A000027093006Z  DS CNT'
'T3 K040    001    -----'
'T3 K040    041    -----'
'T3 K040    081    -----'
'T3 K012    121    -----'
'T4 K016    085    TOTAL DSCBS VTOC'
'T4 A000029104006Z  DSCBS IN VTOC'
'T5 K005    097    ALLOC'
'T5 A000031111006Z  DSCBS ALLOCATED'
'T6 K040    001    -----'
'T6 K040    041    -----'
'T6 K040    081    -----'
'T6 K012    121    -----'
'T7 K023    091    TOTAL TRACKS PER DEVICE'
'T7 A004033117009Z  DEVICE'
'T8 K005    104    ALLOC'
'T8 A004037124009Z  ALLOC CNT'

```

Report Record Format — (RPTNBMVDIC)

B-FORM MVDICT Report

```

BMVDIC
*
*      Mapping of report record used to print B-FORM of MULTIPLE
*      VOLUME DICTIONARY report with owner/project
*      Note: The RPTDSORG field has
*          1. X'02' for PDS and X'FE' for PDSE
*          2. X'40' for PS and X'FC' for PSE
*
BMVDREC  DS   0F           B FORM OF MVD REPORT (BMVD)
RPTBRDW  DS    2H
RPTBTABL DC   CL10'RPTNBMVDIC'
RPTBOWNR DS   CL20         OWNER
RPTBPROJ DS   CL06         PROJECT NUMBER
RPTBINFO DS   0CL86
RPTDSNAM DS   CL44         DATA SET NAME
RPTVOLNO DS   CL6          VOLUME SERIAL
RPTDEVCL DS   XL2          DEVICE CLASS
RPTDSORG DS   XL2          DSORG
RPTRECFM DS   XL1          RECFM
RPTBLKSI DS   XL2          BLOCKSIZE
RPTLRECL DS   XL2          LRECL
RPTSCALO DS   XL1          SECONDARY ALLOCATION
RPTSCQTY DS   XL3          AMOUNT OF SECONDARY ALLOCATION
RPTTKSUM DS   XL3          TOTAL TRACKS ALLOCATED
RPTCREDT DS   XL3          CREATION DATE
RPTEXPDT DS   XL3          EXPIRATION DATE
RPTDSIND DS   XL1          SECURITY
RPTPRMAL DS   XL2          PRIMARY ALLOCATION
RPTIDLET DS   XL2          IDLE TRACKS
RPTVOLAD DS   CL4          VOLUME ADDRESS
RPTCATLG DS   CL1          CATALOG STATUS
RPTLUSDT DS   XL3          LAST USED DATE
RPTNOEXT DS   XL1          NUMBER OF EXTENTS

```

Table Used to Print MVDICT Report

MEMBER = RPTNBMVD

```

'C1 058051001132116020'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 130'
'H1 K032 032 M U L T I P L E V O L U M E '
'H1 K038 064 D A T A S E T D I C T I O N A R Y'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K005 001 OWNER'
'H4 C020001008 OWNER'
'H5 K006 034 DEVICE'
'H5 K004 052 DATA'
'H5 K029 081 --ALLOCATION-- --TRACKS-- '
'H6 K013 001 DATA SET NAME'
'H6 K035 027 VOLUME TYPE PROJ CAT PROT DSORG'
'H6 K029 063 RECFM BLKSZ LRECL TYPE PRIM'
'H6 K016 094 SEC ALLOC IDLE'
'H6 K021 112 EXT CREDIT EXPDT USED'T'
'H7 K040 001 _____ '
'H7 K040 041 _____ '
'H7 K040 081 _____ '
'H7 K012 121 ____ '
'D1 C044027001025 DSN'
'D1 C006071027 VOL'
'D1 C001108049 CAT'
'D1 E001099053 SEC'
'D1 R002077034 DEVICE TYPE'
'D1 C006021041 PROJECT NUMBER'
'D1 V002079058 DSORG'
'D1 W001081064 RECFM'
'D1 H002082069005Z BLKSIZE'
'D1 H002084075005Z LRECL'
'D1 U001086082003 ALLO TYPE'
'D1 H002100086005Z ALLO PRI'
'D1 B003087092005Z ALLO SEC'
'D1 B003090099005Z ALLO SUM'
'D1 H002102105005Z ALLO IDL'
'D1 B001112112003Z EXT NO'
'D1 Y002093116 CREATE'
'D1 Y002096122 Z EXPIRE'

```

```
'D1 Y003109128    Z  LAST USED'  
'T1 K040    001    _____'  
'T1 K040    041    _____'  
'T1 K040    081    _____'  
'T1 K012    121    _____'  
'T2 K015    010    TOTAL DATA SETS '  
'T2 A000000026007Z  COUNTER '  
'T2 K022    072    TOTAL ALLOCATED TRACKS '  
'T2 A000091095009Z  ACCUMULATOR '  
'T3 K017    072    TOTAL IDLE TRACKS '  
'T3 A000102101009Z  ACCUMULATOR '
```


Report Record Format — (RPTNDEVDSB)

Billing Device Report

```

DEVDSB
*
*      Mapping of report record used to print BILLING DEVICE
*      report
*
DEVREC   DS   0F
          DC   H'84,0',CL10'RPTNDEVDSB'
DEVENTRY DS   0CL56
DEVCLASS DS   XL2           DEVICE CLASS
DEVOLCNT DS   H             VOLUME COUNT
DEVDESCNT DS  F             DATA SET COUNT
DEVACCUM DS   F             ACCUMULATED AMOUNT FOR DEVICE
DEVTOTAL DS   F             TOTAL $ AMOUNT FOR DEVICE
DEVDESCR DC   CL40' '       DESCRIPTION
DEVUNITS DC   CL9'K/B DAYS ' or CL9'TRK DAYS '
DSBRELS DC   CL6' '         RELEASE 9.9 = C' 9.9'

```

Table Used for Printing Billing Device Report

```

MEMBER = RPTNDEV
'C1 058053001132'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K010 123 '
'H1 K011 071 PAGE '
'H1 N004 079'
'H1 K042 028 DASD BILLING BY DEVICE TYPE
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K013 120 '
'H2 K006 070 SAMS:Disk'
'H2 C006066076 RELEASE NO'
'H3 K001 001132'
'H4 K011 001 DEVICE TYPE'
'H4 K007 035 VOLUMES'
'H4 K009 045 DATA SETS'
'H4 C009057058 K/B TRK M/B'
'H4 K012 070 TOTAL CHARGE'
'H5 K040 001 _____'
'H5 K041 041 - _____
'D1 C040017001030 DESCRIPTION'
'D1 H002003034005Z VOLUME CNT'
'D1 F004005047005Z DATA SET CNT'
'D1 F004009056010Z K/B TRK M/H'
'D1 $004013069011Z TOTAL CHR'
'T1 K040 001 _____'
'T1 K041 041 _____
'T2 K013 001 TOTAL DEVICES'
'T2 A000 015004Z TOTAL DEVICES'
'T2 K013 022 TOTAL VOLUMES'
'T2 A002003035004Z DSN CNT'
'T3 K015 022 TOTAL DATA SETS'
'T3 A004005046006Z DSN CNT'
'T4 K005 022 TOTAL'
'T4 C008057028 K/B TRK'
'T4 A004009055011Z K/B CNT'
'T5 K013 022 TOTAL CHARGES'
'T5 A004013070012$ CHRG CNT'

```

Report Record Format — (RPTNDISTRS)

Volume Distribution Summary Report

```

DISTRS
*
*      Mapping of report record used to print VOLUME
*      DISTRIBUTION report
*
DETREC   DS      0F
DETCONLN DS      H      RECORD LENGTH, 74 BYTES
          DS      H      (FILLER, ZEROS)
DETTABLE DS     CL10     REPORT NAME, 'RPTNDISTRS'
DETTYPE  DS     XL2      DEVICE TYPE
DETSERAL DS     CL6      VOLUME SERIAL
DETDSCB0 DS      H      DSCB 0 COUNT
DETLT1   DS      H      COUNT OF DATA SETS OF   1 TRACK
DETLT51  DS      H      COUNT OF DATA SETS LT   51 TRACKS
DETLT126 DS      H      COUNT OF DATA SETS LT 126 TRACKS
DETLT301 DS      H      COUNT OF DATA SETS LT 301 TRACKS
DETTGT300 DS      H      COUNT OF DATA SETS GT 300 TRACKS
          DS      H      (FILLER)
DETDEVTK DS     XL4      TRACKS PER DEVICE
DETTKALC DS     XL4      TRACKS ALLOCATED
DETTKAVL DS     XL4      TRACKS AVAILABLE
DETFREXT DS     XL2      FREE EXTENTS
DETTKIDL DS     XL4      TRACKS IDLE
DETFRSPL DS     XL4      1ST LARGEST FREE SPACE
DETFRSPS DS     XL4      2ND LARGEST FREE SPACE
DETFRSPT DS     XL4      3RD LARGEST FREE SPACE
DETVOLAD DS     CL4      VOLUME ADDRESS
DETSMS   DS     CL4      SMS VOLUME STATUS
DETRECLN EQU   * DETREC  LENGTH OF RECORD, 74 BYTES
*

```

Table Used for Printing DISTRS Report

```

MEMBER = RPTNDIST
'C1 058052001132'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K040 039 V O L U M E   D I S T R I B U T I O N   '
'H1 K013 079 S U M M A R Y'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K015 008 DEVICE UNIT SMS'
'H4 K004 026 FREE'
'H4 K035 033 --DISTRIBUTION BY SIZE OF DATA SET--'
'H4 K033 071 -----TRACKS-----'
'H4 K027 106 -----FREE SPACE-----'
'H5 K006 001 VOLUME'
'H5 K015 009 TYPE ADDR STAT'
'H5 K005 026 DSCBS'
'H5 K035 033 ZERO 1-50 51-125 126-300 301->'
'H5 K033 071 DEVICE ALLOC AVAIL IDLE'
'H5 K023 106 EXTENTS THREE LARGEST'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 -----'
'H6 K012 121 -----'
'D1 R006001008 DEV TYPE'
'D1 C006003001 SER'
'D1 C004053015 ADDR'
'D1 C004057020 SMS STAT'
'D1 H002009026005Z BLANK DSCBS'
'D1 H002011032005Z 0'
'D1 H002013040005Z 1-50'
'D1 H002015048005Z 51-125'
'D1 H002017056005Z 126-300'
'D1 H002019063005Z GR 300'
'D1 F004023071006Z TRK DEV'
'D1 F004027080006Z TRK ALLOC'
'D1 F004031089006Z TRK AVAIL'
'D1 F004037098006Z IDLE'
'D1 H002035106005Z EXT'
'D1 F004041113006Z L'
'D1 F004045120006Z S'
'D1 F004049127006Z T'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K010 001 TOTAL VOLS'
'T2 A000000012004Z VOL CNT'

```

```
'T2 A000009024007Z  BLANK DSCB CNT'  
'T2 A000011031006Z  ZERO CNT'  
'T2 A000013038007Z  TO 50 CNT'  
'T2 A000015046007Z  TO 125 CNT'  
'T2 A000017054007Z  TO 300 CNT'  
'T2 A000019062006Z  GT 300 CNT'  
'T2 A004023068009Z  TRKS TOTAL'  
'T2 A004027077009Z  TRKS ALLOC'  
'T2 A004031086009Z  TRKS AVAIL'  
'T2 A004037095009Z  TRKS IDLE'  
'T2 A000035104007Z  EXT CNT'
```

Report Record Format — (RPTNDSNDSB)

Billing by User Code

```

DSNDSB
*
* Mapping of report record used to print BILLING BY USER CODE
*
DETREC   DS   0F
          DC   H'159,0',CL10'RPTNDSNDSB'
DETDSBLR DS   0CL86
DETUSRCD DS   CL12      USER CODE
DETVOLNO DS   CL6       VOLUME SERIAL
DETDNAM  DS   CL44      DATA SET NAME
DETACCUM DS   F         ACCUMULATE AMOUNT
DETNOSCN DS   H         NUMBER OF SCANS
DETCREDT DS   XL3       CREATION DATE
DETCNDT  DS   XL3       LAST SCAN DATE
DETUSEDT DS   XL3       LAST USED DATE
DETFLAGS DS   XL1       FLAGS
DETDEVCL DS   XL2       DEVICE CLASS
DETALLOC DS   H         CURRENT TRACKS ALLOCATED
DETMAXUS DS   H         LARGEST NUMBER OF TRKS ALLOCATED
          DS   H
DETRATE  DS   F         BILLING RATE
DETCOST  DS   F         COST
DETNOSCR DS   H         NUMBER OF TIMES SCRATCHED
DETUSRDS DS   CL40      USER DESCRIPTION
DETUNITS DS   CL9       BILLING UNITS

```

Table Used for Printing Billing by User Code

MEMBER = RPTNDSND

```

'C1 058050001132147012'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K049 043 D A S D B I L L I N G B Y U S E R
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K011 001 USER CODE - '
'H4 C012001013 CODE '
'H4 K013 030 DESCRIPTION - '
'H4 C040097044 DESCRIPTION'
'H5 K001 001132'
'H6 K006 043 DEVICE '
'H6 K015 081 ---TRACKS--- '
'H7 K013 001 DATA SET NAME '
'H7 K026 033 VOLUME TYPE CREDATE '
'H7 K026 060 USEDATE LAST SCAN ALLOC '
'H7 K008 088 MAX USED'
'H7 K016 108 RATE '
'H7 C009137097 K/B TRK'
'H7 K006 127 AMOUNT'
'H8 K040 001 _____ '
'H8 K040 041 _____ '
'H8 K040 081 --- _____ '
'H8 K012 121 -- _____'
'D1 C044019001028 DSN'
'D1 C006013033 VOL'
'D1 R002079043 DEVICE'
'D1 J003069051 Z CREATE'
'D1 J003075060 Z LAST USE'
'D1 J003072070 Z LAST SCAN'
'D1 H002081081005Z TRKS ALLOCATED'
'D1 H002083091005Z MAX USED'
'D1 F004063097009Z K/B TRK M/H'
'D1 F004087107006 RATE'
'D1 $004091123010 AMOUNT'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K015 005 TOTAL DATA SETS'
'T2 A000000021005Z DSN CNT'
'T2 K022 057 TOTAL ALLOCATED TRACKS'
'T2 A000081080010Z ALLOCATED TRK CNT'
'T3 K005 057 TOTAL'
'T3 C008138063 K/B TRK'
'T3 A004063094012Z K/B CNT'

```

```
'T4 K012    057    TOTAL AMOUNT'  
'T4 A004091121012$ AMT CNT'
```


Report Record Layout — (RPTNDSNX)

```

'C   058051001127128'
'H113132   001   STANDARD HEADER - LINE 1'
'H1 K041   045   X C O P Y   D A T A   S E T   S T A T U S'
'H1 K007   122   PAGE '
'H1 N004   129   PAGE NO'
'H2 4132   001   STANDARD HEADER - LINE 2'
'H3 K001   001132 '
'H4 K025   054   -----OLD-----   KILO'
'H4 K005   084   BLOCK'
'H4 K006   099   ACTION'
'H4 K024   108   -----NEW-----   OUTPUT'
'H5 K013   001   DATA SET NAME'
'H5 K036   045   ARCDATE   KEY   FILE   BLOCK   BYTES '
'H5 K020   084   COUNT EXPDATE   TAKEN'
'H5 K024   108   KEY   FILE   BLOCK   DD'
'H7 K040   001   -----
'H7 K040   041   --- -----
'H7 K040   081   -----
'H7 K011   121   -----
'H607008115001   SIMULATE'
'D1 C044003001038   DSNAME'
'D1 J003057045   ARCDT'
'D1 C006085054   VOLSER'
'D1 H002073060005Z   FILE NO'
'D1 F004075065007Z   BLOCK'
'D1 F004079073007Z   BYTES'
'D1 H002083082005Z   BLOCKS'
'D1 J003060090   EXPDT'
'D1 C008093099   ACTION'
'D1 C006101108   VOLSER'
'D1 H002111114005Z   FILE NO'
'D1 F004107119007Z   BLOCK'
'D1 C001113131   DEVICE'

```

Record Report Format

```
*
* REPORT RECORD FOR COPY DSNINDEX PROCESSING
*
          DS      0F
RPTREC   DS      0CL136
RPTLEN   DS      H,H
RPTNAME  DS      CL12          RPTDSNXCP
RPTLRDSN DS      CL90          USE ONLY THE FIRST 90 BYTES
DISPOSIT DS      CL8
RPTARKEY DS      CL6
RPTBLKNO DS      F
RPTFILNO DS      H
RPTTUNIT DS      AL1
          DS      X
RPTSIM   DS      CL8
```

Report Record Layout — (RPTNVOLX)

```

'C 058048001132150'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K037 047 X C O P Y V O L U M E S T A T U S'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132 '
'H4 K036 052 -----ORIGINAL-----'
'H4 K027 106 -----COPIED-----'
'H5 K011 001 ARCHIV VOL'
'H5 K004 061 DATA'
'H5 K005 069 BLOCK'
'H5 K004 077 KILO'
'H5 K002 085 NO'
'H5 K004 106 DATA'
'H5 K005 114 BLOCK'
'H5 K004 122 KILO'
'H5 K002 130 NO'
'H6 K026 001 KEY CNT DATA SET NAME'
'H6 K007 052 EXPDATE'
'H6 K042 061 SETS COUNT BYTES FEET ACTION TAKEN'
'H6 K027 106 SETS COUNT BYTES FEET'
'H7 K040 001 -----'
'H7 K040 041 -----'
'H7 K040 081 -----'
'H7 K012 121 -----'
'H807008137001 SIMULATE'
'D1 C006003001 VOLSER'
'D1 B001099010001 VOLCOUNT'
'D1 C044033014037 VOLDSNAM'
'D1 J003094052 VOLEXPDT'
'D1 H002077060005Z VOLDSCNT'
'D1 F004079066007Z VOLBKCNT'
'D1 F004083074007Z VOLBYCNT'
'D1 H002097082005Z VOLFEET'
'D1 H002117105005Z NEWDSCNT'
'D1 F004119112007Z NEWBKCNT'
'D1 F004123120007Z NEWBYCNT'
'D1 H002115128005Z NEWFEET'
'D1 C008127091 ACTION TAKEN'

```

Report Record Format

```
*
** XCOPY VOLUME REPORT RECORD
*
RPTREC    DS      0F
           DS      H,H
RPTNAME   DS      CL12
RPTLRVOL  DS      CL110
           DS      H
UNEXFEET  DS      H
UNEXFILE  DS      H
UNEXBLKS  DS      F
UNEXBYTS  DS      F
DISPOSIT  DS      CL8
           DS      H
RPTSIM    DS      CL8
RPTLENTN  EQU   *-RPTREC
```

Report Record Format — (RPTNDSINDEX)

Data Set Index Report

```

DSINDEX
*
*      Mapping of report record used to print DATA SET INDEX
*      Report
RPTBUILD DS    0F
          DC    H'86,0'
RPTTABLE DC    CL10'RPTNDSINDEX'
RPTDSNAM DC    CL44          DATA SET NAME
RPTVOLNO DC    CL6          VOLUME SERIAL
RPTDEVCL DC    XL2          DEVICE CLASS
RPTDSORG DC    XL2          DSORG
RPTRECFM DC    XL1          RECFM
RPTBLKSI DC    XL2          BLOCKSIZE
RPTLRECL DC    XL2          LRECL
RPTSCALO DC    XL1          ALLOCATION
RPTSCQTY DC    XL3          SECONDARY QUANTITY
RPTTKSUM DC    XL3          TOTAL TRACKS ALLOCATED
RPTCREDT DC    XL3          CREATION DATE
RPTEXPDT DC    XL3          EXPIRATION DATE

```

Table Used for Printing Data Set Index Report

```

MEMBER = RPTNDSIN
'C1  058053003045086'
'H113132  001  STANDARD HEADER - LINE 1'
'H1 K007  122  PAGE '
'H1 N004  130  PAGE NO'
'H1 K027  053  DATA SET INDEX'
'H2 4132  001  STANDARD HEADER - LINE 2'
'H3 K001  001132'
'HC K013  001  DATA SET NAME'
'HC K006  036  VOLUME'
'H4 K045  001  _____
'H4 K045  046  _____
'H4 K045  091  _____
'D1 C044001001033  DSN'
'D1 C006045036  VOL'
'T1 K040  001  _____'
'T1 K040  041  _____'
'T1 K040  081  _____'
'T1 K012  121  _____'
'T2 K015  015  TOTAL DATA SETS'
'T2 A000000031009Z  DSN CNT'

```

Report Record Format — (RPTNDSUTIL)

Data Set Utilization Report

```

DSUTIL
*
*   Mapping of report record used to print DATA SET UTILIZATION
*   report
RPTBUILD DS    0F
          DC    H'127,0'
RPTTABLE DC    CL10'RPTNDSUTIL'
RPTDSNCF DC    CL8' ' '          1
RPTSQFLD DC    XL3'000000'        9
RPTDSN  DC    CL44' ' '          12
RPTVOL  DC    CL6' ' '           56
RPTCDATE DC    XL3'000000'        62
RPTEDATE DC    XL3'000000'        65
RPTOPENS DC    XL4'00000000'      68
RPTJOBNM DC    CL8' ' '          72
RPTTRKS DC    XL2'0000'           80
RPTUSED DC    XL2'0000'           82
RPTIDLE DC    XL2'0000'           84
RPTLUPDT DC    XL3'000000'        86
RPTLUSED DC    XL3'000000'        89
RPTLUFLG DC    C' ' '            92
RPTCATLG DC    C' ' '            93
RPTSECTY DC    C' ' '            94
RPTOWNER DC    CL11' ' '         95

```

Table Used for Printing DSUTIL Report

```

MEMBER = RPTNDSUT
'C1  058052001132116008'
'H113132  001  STANDARD HEADER - LINE 1'
'H1 K007  122  PAGE '
'H1 N004  129  PAGE NO'
'H1 K039  047  DATA SET UTILIZATION'
'H2 4132  001  STANDARD HEADER - LINE 2'
'H3 K003  080  C P'
'H4 K004  001  FLAG'
'H4 K035  007  -----DATES-----'
'H4 K010  080  A R OPEN'
'H4 K019  114  -----TRACKS-----'
'H5 K004  001  DAYS'
'H5 K035  007  USEDATE MODDATE CREDATE EXPDATE '
'H5 K013  043  DATA SET NAME'
'H5 K017  073  VOLUME T O COUNT'
'H5 K007  098  JOBNAME'

```

```

'H5 K019    114  ALLOC    USED    IDLE'
'H6 K040     01  —————'
'H6 K040     04  - ————— - — -'
'H6 K040     08  - — - ————— ———'
'H6 K012     12  —————'
'D1 C001100003      USE'
'D1 J003089007      USED'T'
'D1 J003086016      MODDT'
'D1 J003062025      CREDIT'
'D1 J003065034      EXPDT'
'D1 C044012043029   DSN'
'D1 C006056073      VOL'
'D1 C001093080      CAT'
'D1 E001094082      SEC'
'D1 F004068085005Z  OPENS'
'D1 C008072098      JOBNAME'
'D1 H002080113006Z  ALLOC'
'D1 H002082120006Z  USED'
'D1 H002084127006Z  IDLE'
'T1 K040     01  —————'
'T1 K040     04  —————'
'T1 K040     08  —————'
'T1 K012     12  —————'
'T2 K015     05  TOTAL DATA SETS'
'T2 A000000021005Z  DSN CNT'
'T2 K018     09  TOTAL TRACKS ALLOC'
'T2 A000080110009Z  ALLOC CNT'
'T3 K043     01  * - DATA SET HAS NOT BEEN USED WITHIN VALUE
'T3 K032     04  SPECIFIED IN FLAGDAYS PARAMETER.'
'T3 K004     10  USED'
'T3 A000082117009Z  USE CNT'
'T4 K041     01  $ - LAST USE DATE IS ZERO; VALUE SHOWN IS
'T4 K014     04  CREATION DATE.'
'T4 K004     10  IDLE'
'T4 A000084124009Z  IDLE CNT'

```

Report Record Format — RPTNDAYWEK

Utilization by Date and Time Report

```

SMFRPT
*
*      Mapping of Report record used to print UTILIZATION BY
*      DATE AND TIME REPORT
*
RECORD   DS   0CL76
DSNAME   DC   CL44' '   DATA SET NAME
VOLSER   DC   CL6' '    VOLUME SERIAL NUMBER
          DC   CL2      BLANKS
JOBTIME  DC   XL4'00'   DURATION OF JOB
JOBDATE  DS   0PL4      DATE (JULIAN) OF JOB RUN
          DC   PL1'0'
          DC   PL3'0'
SMFTIME  DC   XL4'00'   SMF JOB TIME
SMFDATE  DC   PL4       SMF (JULIAN) DATE OF JOB RUN
          DC   X'0'
START    DC   XL3'00'   START DATE
          DC   X'0'
STOP     DC   XL3'00'   STOP DATE

```

Table Used for Printing DAYWEK Report

```

MEMBER = RPTNDAYW
'C1 058051'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K055 037 UTILIZATION BY DATE AND ',
'TIME'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K011 064 DAY OF WEEK'
'H4 K021 098 HOUR OF DAY'
'H5 K027 105 1 1 1 1 1 1 1 1 1 2 2 2 2'
'H6 K013 001 DATA SET NAME'
'H6 K006 046 VOLUME'
'H6 K019 060 S M T W T F S'
'H6 K047 085 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3'
'H7 K040 001 -----'
'H7 K040 041 -----'
'H7 K040 081 -----'
'H7 K012 121 -----'
'D1 C044001001 DSN'
'D1 C006045046 VOL'

```


'D1 C001051060	SUN'
'D1 C001052063	MON'
'D1 C001053066	TUE'
'D1 C001054069	WED'
'D1 C001055072	THR'
'D1 C001056075	FRI'
'D1 C001057078	SAT'
'D1 C001058085	0'
'D1 C001059087	1'
'D1 C001060089	2'
'D1 C001061091	3'
'D1 C001062093	4'
'D1 C001063095	5'
'D1 C001064097	6'
'D1 C001065099	7'
'D1 C001066101	8'
'D1 C001067103	9'
'D1 C001068105	10'
'D1 C001069107	11'
'D1 C001070109	12'
'D1 C001071111	13'
'D1 C001072113	14'
'D1 C001073115	15'
'D1 C001074117	16'
'D1 C001075119	17'
'D1 C001076121	18'
'D1 C001077123	19'
'D1 C001078125	20'
'D1 C001079127	21'
'D1 C001080129	22'
'D1 C001081131	23'

Report Record Format — (RPTNFFREESP)

Volume Freespace Report

```
FREESP
*
* Mapping of report record used to print FREESPACE report
*
DETREC   DS      0F
DETCONLN DS      H      RECORD LENGTH, 61 bytes
          DS      H      (FILLER  ZEROS)
DETTABLE DS    CL10      REPORT NAME 'RPTNFFREESP'
DETDEVCL DS    XL2      DEVICE CLASS
DETVOLNO DS    CL6      VOLSER
DETBEGTK DS    XL4      BEGINNING TRACK
DETENDTK DS    XL4      ENDING TRACK
DETLENTK DS    XL4      TRACK COUNT
DETVOLAD DS    CL4      VOLUME UNIT ADDRESS
DETSMS   DS    CL21      SMS VOLUME STATUS
DETEXTNO DS    XL2      EXTENT NUMBER
DETRECLN EQU  * DETREC  LENGTH OF RECORD, 61 BYTES
```

Table Used to Print Volume Freespace Report

MEMBER = RPTNFREE

```

'C1 058051008017090008'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K047 042 V O L U M E F R E E S P A C E D E T A I L'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K006 001 VOLUME'
'H4 C006003008 VOL NO'
'H4 K006 016 DEVICE'
'H4 R002001023 TYPE'
'H4 K004 031 ADDR'
'H4 C004021036 ADDRESS'
'H4 K015 043 THIS VOLUME IS '
'H4 C021025058021 SMS STATUS '
'H5 K001 001132'
'HC K013 001 TRACK LEN'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 - -----'
'H6 K012 121 -----'
'D1 F004009001006Z TRK'
'D1 F004017008006Z LEN'
'T2 K040 001 -----'
'T2 K040 041 -----'
'T2 K040 081 -----'
'T2 K012 121 -----'
'T3 K001 001132 '
'T4 K017 014 TOTAL FREE TRACKS'
'T4 A004017032009Z FR SP CNT'

```

Report Record Format — (RPTNLAYOUT)

Volume Layout Report

```
LAYOUT
*
*      Mapping of report record used to print VOLUME LAYOUT
*      report
*
DETREC   DS      0F
DETCONLN DS      H      RECORD LENGTH, 106 BYTES
          DS      H      (FILLER, ZEROS)
DETTABLE DS     CL10     REPORT NAME 'RPTNLAYOUT'
DETDEVCL DS     XL2      DEVICE CLASS
DETVOLNO DS     CL6      VOLUME SERIAL
DETBEGTK DS     XL4      BEGINNING TRACK
DETENDTK DS     XL4      ENDING TRACK
DETLENTK DS     XL4      TRACK LENGTH
DETEXTNO DS     HL2      EXTENT NUMBER
DETDSNAM DS     CL44     DATA SET NAME
DETEXTYP DS     CL1      EXTENT TYPE
DETVOLAD DS     CL4      VOLUME ADDRESS
DETSMS   DS     CL21     SMS VOLUME STATUS
DETRECLN EQU  * DETREC  LENGTH OF RECORD, 106 BYTES
*
```

Table Used for Volume Layout Report

MEMBER = PRTNLAYO

```

'C1 060051008017096008'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K026 053 V O L U M E L A Y O U T '
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K006 001 VOLUME'
'H4 C006003008 VOL NO'
'H4 K006 016 DEVICE'
'H4 R002001023 DEVICE'
'H4 K004 031 ADDR'
'H4 C004068036 ADDRESS'
'H4 K014 043 THIS VOLUME IS'
'H4 C021072058021 SMS VOLUME STATUS '
'H5 K015 001 D - DATA EXTENT'
'H5 K019 020 S - SHARED CYLINDER'
'H5 K014 043 F - FREE SPACE'
'H5 K008 061 V - VTOC'
'H6 K001 001132'
'HC K017 001 TYP LEN '
'H7 K040 001 -----'
'H7 K040 041 -- -----'
'H7 K040 081 -----'
'H7 K012 121 -----'
'D1 C001067001 TYPE'
'D1 F004017003006Z LEN'

```

Report Record Format — (RPTNMAPPER)

Volume Mapping Report

```

MAPPER
*
*      Mapping of report record used to print VOLUME MAPPING
*      report
*
DETREC   DS      0F
DETCONLN DS      H      RECORD LENGTH, 110 BYTES
          DS      H      (FILLER, ZEROS)
DETTABLE DS     CL10     REPORT NAME, 'RPTNMAPPER'
DETDEVCL DS     XL2      DEVICE CLASS
DETVOLNO DS     CL6      VOLUME SERIAL
DETBEGTK DS     XL4      BEGINNING TRACK
DETENDTK DS     XL4      ENDING TRACK
DETLENTK DS     XL4      TRACK COUNT
DETEXTNO DS      H      EXTENT NUMBER
DETDSNAM DS     CL44     DATA SET NAME
DETEXTYP DS     CL1      EXTENT TYPE
DETVOLAD DS     CL4      VOLUME ADDRESS
DETCCVAL DS     XL2      CC VALUE
DETHHVAL DS     XL2      HH VALUE
DETSMS   DS     CL21     SMS VOLUME STATUS
DETRECLN EQU   * DETREC  LENGTH OF RECORD, 110 BYTES
*

```

Table Used to Print Volume Mapping Report

MEMBER = RPTNMAPP

```

'C1 058051002066100008'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K024 056 V O L U M E M A P '
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K006 001 VOLUME'
'H4 C006003008 VOL NO'
'H4 K006 016 DEVICE'
'H4 R002001023 DEVICE'
'H4 K004 031 ADDR'
'H4 C004068036 ADDRESS'
'H4 K014 043 THIS VOLUME IS'
'H4 C021076058 SMS STATUS'
'H5 K001 001132'
'HC K013 001 DATA SET NAME'
'HC K002 043 CC'
'HC K002 047 HH'
'HC K005 051 TRACK'
'HC K003 059 LEN'
'HC K003 063 EXT'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 -----'
'H6 K011 121 -----'
'D1 C044023001040 DSN'
'D1 X002072042004 CC'
'D1 X001075047004 HH'
'D1 F004009050006Z TRK'
'D1 F004017057006Z LEN'
'D1 H002021063003Z EXT NO'

```

Report Record Format — (RPTNMEMBER)

PDS Directory / Member Report

```
MEMBER
*
*      Mapping of report record used to print MEMBER report
*
MEMREC   DS    0F
          DC    H'74'
          DS    H
MEMTBLNM DC    CL10'RPTNMEMBER'
MEMDSN   DS    CL44          DATA SET NAME
MEMVOL   DS    CL6           VOLUME
MEMNAME  DS    CL8           MEMBER NAME
MEMALIAS DS    C             ALIAS
MEMRENT  DS    C             RE-ENTRANT
```


Table Used to Print MEMBER Report

```

MEMBER = RPTNMEMB
'C1 058051010013066050'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K059 036 P A R T I T I O N E D   D A T A   S E T   D I R E',
' C T O R Y'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K013 001 DATA SET NAME'
'H4 C044001016 DSN'
'H4 K006 062 VOLUME'
'H4 C006045069 VOL'
'H4 K006 077 DEVICE'
'H4 R002061084 DEVICE'
'H4 K004 092 ADDR'
'H4 C004063098 ADDRESS'
'H5 K001 001132'
'HC K006 002 MEMBER'
'H7 K040 001 -----'
'H7 K040 041 -----'
'H7 K040 081 -----'
'H7 K012 121 -----'
'D1 C008051001 MBR'
'D1 C001059010 ALIAS'
'T2 K040 001 -----'
'T2 K040 041 -----'
'T2 K040 081 -----'
'T2 K012 121 -----'
'T4 K013 015 TOTAL MEMBERS'
'T4 A000000029006Z MBR CNT'

```

Report Record Format — (RPTNMVDICT)

Multiple Volume Data Set Dictionary Report

```

MCDICT
*
*      Mapping of report record used to print MULTIPLE VOLUME
*      DATA SET DICTIONARY report
*
*
RPTBUILD DS      0F
          DC      H'100,0'
RPTTABLE DC      CL10'RPTNMVDICT'
RPTAINFO DS      0CL86
RPTDSNAM DS      CL44          DATA SET NAME
RPTVOLNO DS      CL6          VOLUME SERIAL
RPTDEVCL DS      XL2          DEVICE CLASS
RPTDSORG DS      XL2          DSORG
RPTRECFM DS      XL1          RECFM
RPTBLKSI DS      XL2          BLOCKSIZE
RPTLRECL DS      XL2          LRECL
RPTSCALO DS      XL1          SECONDARY ALLOCATION, (TRK CYL)
RPTSCQTY DS      XL3          AMOUNT OF SECONDARY ALLOCATION
RPTTKSUM DS      XL3          TOTAL TRACKS ALLOCATED
RPTCREDT DS      XL3          CREATION DATE
RPTEXPDT DS      XL3          EXPIRATION DATE
RPTDSIND DS      XL1          SECURITY
RPTPRMAL DS      XL2          PRIMARY ALLOCATION
RPTIDLET DS      XL2          IDLE TRACKS
RPTVOLAD DS      CL4          VOLUME ADDRESS
RPTCATLG DS      CL1          CATALOG STATUS
RPTLUSDT DS      XL3          LAST USED DATE
RPTNOEXT DS      XL1          NUMBER OF EXTENTS

```

Table Used to Print MVDICT Report

```

MEMBER = RPTNMVDI
'C1  058052001132090'
'H113132  001'
'H1 K007  122      PAGE '
'H1 N004  129'
'H1 K032  033  M U L T I P L E  V O L U M E  '
'H1 K038  065  D A T A  S E T  D I C T I O N A R Y'
'H2 4132  001'
'H3 K003  048  C P'
'H4 K015  036  DEVICE UNIT A R'
'H4 K031  074  ---ALLOCATION-- ----TRACKS-----'
'H5 K013  001  DATA SET NAME'

```

```

'H5 K006 029 VOLUME'
'H5 K011 036 TYPE ADDR'
'H5 K007 048 T O DSO'
'H5 K026 056 RECFM BLKSZ LRECL TYP PRIM'
'H5 K016 084 SEC ALLOC IDLE'
'H5 K030 102 EXT CREDATE EXPDATE USEDATE '
'H6 K040 001 -----'
'H6 K040 041 - ---- - - - - - - - - - - - - - - - -'
'H6 K040 081 -----'
'H6 K012 121 -----'
'D1 C044001001027 DSN'
'D1 C006045029 VOL'
'D1 R002051036 DEVICE TYPE'
'D1 C004078043 ADDRESS'
'D1 C001082048 CAT'
'D1 E001073050 SEC'
'D1 V002053052 DSORG'
'D1 W001055056 RECFM'
'D1 H002056062005Z BLKSIZE'
'D1 H002058068005Z LRECL'
'D1 U001060074003 ALLO TYPE'
'D1 H002074078005Z ALLO PRI'
'D1 B003061084005Z ALLO SEC'
'D1 B003064090005Z ALLO SUM'
'D1 H002076096005Z ALLO IDL'
'D1 B001086102003Z EXT NO'
'D1 J002067106 Z CREATE'
'D1 J002070115 Z EXPIRE'
'D1 J003083124 Z LAST USED'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K015 010 TOTAL DATA SETS'
'T2 A000000026007Z COUNTER'
'T2 K022 063 TOTAL ALLOCATED TRACKS'
'T2 A000065086009Z ACCUMULATOR'
'T3 K017 063 TOTAL IDLE TRACKS'
'T3 A000076092009Z ACCUMULATOR'

```

Report Record Format — (RPTNPDSTAT)

Partitioned Data Set Status Report

```

PDSTAT
*
*      Mapping of report record used to print PARTITIONED DATA
*      SET STATUS report
*
DETPDSTB DS    0F
          DC    H'92,0'          record length
DETTABLE DC    CL10'RPTNPDSTAT'
DETDSTNAM DS    CL44          DATA SET NAME
DETDEVIC DS    XL2          DEVICE CLASS
DETVOLSR DS    CL6          VOLSER
DETEXTNT DS    XL1          NUMBER OF EXTENTS
DETTTRKAL DS    XL2          TRACKS REQUESTED WHEN CREATED
DETTTRKUS DS    XL2          TRACKS USED
DETTTRKID DS    XL2          TRACKS IDLE
DETDIRAL DS    XL2          DIRECTORY BLOCKS ALLOCATED
DETDIRUS DS    XL2          DIRECTORY BLOCKS USED
DETDIREM DS    XL2          DIRECTORY BLOCKS EMPTY
DETMBCU DS    XL2          MEMBER COUNT
DETMBRPO DS    XL2          MEMBER LIMIT
DETMBRGR DS    XL2          AVAILABLE ENTRIES
DETMLEN DS    XL2          ENTRY LENGTH
DETALPRI DS    XL2          TRACKS ALLOCATED
DETALSEC DS    XL2          SECONDARY ALLOCATION
DETALTYP DS    XL1          ALLOCATION TYPE

```

Table Used to Print PDS Status Report

```

MEMBER = RPTNPDST
'C1 058052001132092'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 130'
'H1 K052 040 P A R T I T I O N E D   D A T A   S E T
S T A T U S'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'H4 K006 038 DEVICE'
'H4 K016 045 -SPACE REQUEST-'
'H4 K017 063 ---TRACKS---'
'H4 K018 089 -DIRECTORY BLOCKS-'
'H4 K022 111 -DIRECTORY ENTRIES--'
'H5 K013 001 DATA SET NAME'
'H5 K006 030 VOLUME'

```

```

'H5 K004    039    TYPE'
'H5 K004    045    TYPE'
'H5 K004    051    PRIM' '
'H5 K003    058    SEC'
'H5 K005    063    ALLOC'
'H5 K004    070    USED'
'H5 K004    076    IDLE'
'H5 K003    083    EXT'
'H5 K005    089    ALLOC'
'H5 K004    096    USED'
'H5 K004    103    IDLE'
'H5 K005    111    ALLOC'
'H5 K004    118    USED'
'H5 K004    124    IDLE'
'H5 K003    130    LEN'
'H6 K040    001    _____ -- --'
'H6 K040    041    -- _____ '
'H6 K040    081    -- _____ '
'H6 K012    121    _____'
'D1 C044001001      DSNAME'
'D1 C006047030      VOLSER'
'D1 R006045038      DEVICE'
'D1 U001078046003   ALLOC TYPE'
'D1 H002074050005Z  ALLOC PRIM'
'D1 H002076056005Z  ALLOC SEC'
'D1 H002054063005Z  TRKS ALLOC'
'D1 H002056069005Z  TRKS USED'
'D1 H002058075005Z  TRKS IDLE'
'D1 B001053084002Z  EXTENTS'
'D1 H002060089005Z  DIR BLKS ALLOC'
'D1 H002062095005Z  DIR BLKS USED'
'D1 H002064102005Z  DIR BLKS - IDLE'
'D1 H002068111005Z  MBR LIMIT - ALLOC'
'D1 H002066117005Z  MBR - CURRENT - USED'
'D1 H002070123005Z  MBR - GROWTH - IDLE'
'D1 H002072130003Z  AVE LEN'
'T1 K040    001    _____'
'T1 K040    041    _____'
'T1 K040    081    _____'
'T1 K012    121    _____'
'T2 K015    005    TOTAL DATA SETS'
'T2 A000000021006Z  DSN CNT'
'T2 K018    040    TOTAL TRACKS ALLOC'
'T2 A000054059009Z  ALLOC CNT'
'T3 K004    053    USED'
'T3 A000056065009Z  USE CNT'
'T4 K004    053    IDLE'
'T4 A000058071009Z  IDLE CNT'

```

Report Record Format — (RPTNPOCCHH, RPTNPODUMP, and RPTNPOTTRD)

Partitioned Data Set Directory (CCHHR, DUMP, TTRD)

```
*
*      Mapping of report record used to print the POCCHH,
*      PODUMP, AND POTTRD reports.
*
RECORD   DS    0F
RECLLEN  DS    H           Length of record (86 to 148)
          DS    H
RPTPGM   DS    CL10        Name of module to format the rpt
DSNAME   DS    CL44        Data set name
VOLUME   DS    CL6         Volume
KEY       DS    CL8         Key
CCHHR    DS    XL5         CCHHR of member
ALIAS     DS    C           Member name is an alias
DEVTYPE  DS    XL2         Devtype
VOLAD     DS    CL4         Volume address
#ENTLEN  DS    H           Length of user data
#DIRENT  DS    XL62        User data
```

Report Record Format — (RPTNPOXREF, RPTNPO1REF, and RPTNPO2REF)

Data Set Name Cross-Reference for PDS Member Names

```

*      Mapping of report record used to print the POXREF,
*      PO1REF, and PO2REF reports.
*
*
POXREC   DS    0F
POXLEN   DS    H           Length of record
          DS    H
POXTBLN  DS    CL10        Name of module to format the rpt
POXMBR   DS    CL8         Member name
POXDSN   DS    CL44        Data set name
POXVOL   DS    CL6         Volume serial
POXALIAS DS    C           Member name is an alias
POXDEV   DS    XL2         Device type
POXVADDR DS    CL4         Volume address

I          RACF PROFILE TO ARCHIVE DSNAME X-REF

```

Report Record Format — (RPTNRACFDN)

RACF Profile X-REF in Dsname Sequence

```

RPTNRACFDN
*
*      Mapping of report record used to print RACF PROFILE TO
*      ARCHIVE DSNAME X-REF, in archive dsname/date/time sequence
*
RDNRECD  DC    H'106,0'
          DC    CL10'RPTNRACFDN'
RDNDSN   DC    CL44' '      Data set name
RDNDATE  DC    XL3'0'       Archive date
RDNTIME  DC    CL4' '       Archive time
RDNPROF  DC    CL41' '      Profile name

```

Table Used to Print X-REF Report

MEMBER = RPTNRFDN

```
'C1 058052001132104'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K045 030 R A C F P R O F I L E T O A R C H I V E'
'H1 K026 075 D S N A M E X - R E F'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K033 049 ARCHIVE DSNAME/DATE/TIME SEQUENCE'
'H3 K001 001132'
'H4 K007 001 ARCHIVE'
'H4 K007 049 ARCHIVE'
'H4 K007 062 PROFILE'
'H5 K013 001 DATA SET NAME'
'H5 K010 047 TIME DATE'
'H5 K013 062 DATA SET NAME'
'H6 K040 001 _____'
'H6 K040 041 --- _____'
'H6 K022 081 _____'
'D1 C044001001 DSN'
'D1 C004048047 ARCHIVE TIME'
'D1 J003045052 Z ARCHIVE DATE'
'D1 C041052062 PROFILE NAME'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K015 005 TOTAL DATA SETS'
'T2 A000000021005Z DSN CNT'
```


Report Record Format — (RPTNRACFPF)

RACF Profile X-REF in Profile Name Sequence

```

RPTNRACFPF
*
*   Mapping of report record used to print RACF PROFILE TO
*   ARCHIVE DSNAME X-REF, in profile name sequence
*
RPFRECD  DC    H'106,0'
          DC    CL10'RPTNRACFPF'
RPFPROF  DC    CL41' '           Profile name
RPFDSN   DC    CL44' '           Data set name
RPFDATE  DC    XL3'0'            Archive date
RPFTIME  DC    CL4' '            Archive time

```

Table Used to Print X-REF Report

MEMBER = RPTNRFPF

```
'C1 058052001132104'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 130 PAGE NO'
'H1 K045 030 R A C F P R O F I L E T O A R C H I V E'
'H1 K026 075 D S N A M E X - R E F'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K021 055 PROFILE NAME SEQUENCE'
'H3 K001 001132'
'H4 K007 001 PROFILE'
'H4 K007 047 ARCHIVE'
'H4 K007 097 ARCHIVE'
'H5 K013 001 DATA SET NAME'
'H5 K013 047 DATA SET NAME'
'H5 K010 095 TIME DATE'
'H6 K040 001 _____'
'H6 K040 041 - _____'
'H6 K024 081 _____'
'D1 C041001001 PROFILE NAME'
'D1 C044042047 DSN'
'D1 C004089095 ARCHIVE TIME'
'D1 Y003086100005Z ARCHIVE DATE'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K015 005 TOTAL DATA SETS'
'T2 A000000021005Z DSN CNT'
```

Report Record Format — (RPTNSVDICT)

Single Volume Dictionary Report

```

SVDICT
*
*      Mapping of report record used to print SINGLE VOLUME
*      DICTIONARY report
*      Note: The RPTDSORG field has:
*              1. X'02' for PDS and X'FE' for PDSE
*              2. X'40' for PS and X'FC' for PSE
*
RPTBUILD DS    0F
           DC    H'100,0'
RPTTABLE DC    CL10'RPTNSVDICT'
RPTDEVCL DS    XL2          DEVICE CLASS
RPTVOLNO DS    CL6          VOLUME SERIAL
RPTDSNAM DS    CL44         DATA SET NAME
RPTDSORG DS    XL2          DSORG
RPTRECFM DS    XL1          RECFM
RPTBLKSI DS    XL2          BLOCKSIZE
RPTLRECL DS    XL2          LRECL
RPTSCALO DS    XL1          SECONDARY ALLOCATION
RPTSCQTY DS    XL3          SECONDARY ALLOCATION QUANTITY
RPTTKSUM DS    XL3          TOTAL TRACKS ALLOCATED
RPTCREDT DS    XL3          CREATION DATE - YYDDD
RPTEXPDT DS    XL3          EXPIRATION DATE - YYDDD
RPTDSIND DS    XL1          SECURITY
RPTPRMAL DS    XL2          PRIMARY ALLOCATION
RPTIDLET DS    XL2          IDLE TRACKS
RPTVOLAD DS    CL4          VOLUME ADDRESS
RPTCATLG DS    CL1          CATALOG STATUS
RPTLUSDT DS    XL3          LAST USED DATE - YYDDD
RPTNOEXT DS    XL1          EXTENT COUNT

```

Table Used to Print SVDICT Report

```

'MEMBER = RPTNSVDI
'C1  058052001132096008'
'H113132  001'
'H1 K007  122      PAGE '
'H1 N004  129'
'H1 K028  034  S I N G L E  V O L U M E  '
'H1 K038  062  D A T A  S E T  D I C T I O N A R Y'
'H2 4132  001'
'H3 K003  048  C P'
'H4 K015  036  DEVICE UNIT A R'
'H4 K031  074  ---ALLOCATION-- ----TRACKS-----'
'H5 K013  001  DATA SET NAME'

```


Report Record Format — (RPTNVOLDSB)

Billing Volume Report

```

VOLDSB
*
*      Mapping of report record used to print BILLING VOLUME
*      report
*
VOLREC   DS   0F
         DC   H'76,0',CL10'RPTNVOLDSB'
VOLENTRY DS   0CL24
VOLSERNO DS   CL6           VOLSER
VOLDSCNT DS   H           DATA SET COUNT
VOLACCUM DS   F           ACCUMULATED AMOUNT FOR VOLUME
VOLTOTAL DS   F           VOLUME TOTAL - $
VOLDEVCL DS   XL2          DEVICE CLASS
VOLNOSCN DS   H           NUMBER OF SCANS
VOLSCNDT DS   XL3          SCAN DATE - YY DDDD
         DS   XL1          RESERVED
VOLUNITS DC   CL9'K/B DAYS ' or CL9'TRK DAYS '
VOLRELSE DC   CL6' '       RELEASE 9.9 = C' 9.9'
         DC   CL23' '       FILLER FOR SORT MINIMUM LENGTH

```

Table Used for Printing Billing Volume Report

```

MEMBER = RPTNVOLD
'C1 058053001132'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K010 123 '
'H1 K011 071 PAGE '
'H1 N004 078'
'H1 K046 025 D A S D B I L L I N G B Y V O L U M E'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K013 120 '
'H2 K006 069 SAMS:Disk'
'H2 C006034075 RELEASE NO'
'H3 K001 001132'
'H4 K028 001 VOLUME DEVTYPE NO SCANS'
'H4 K009 033 LAST SCAN'
'H4 K009 044 DATA SETS'
'H4 C009025056 K/B TRK M/B'
'H4 K012 069 TOTAL CHARGE'
'H5 K040 001 _____'
'H5 K040 041 - _____'
'D1 C006001001 VOL'
'D1 R002017011 DEV CLASS'
'D1 H002019022005Z NO SCANS'
'D1 Y003021035 SCAN DATE'
'D1 H002007045005Z DATA SET CNT'
'D1 F004009054010Z K/B TRK M/H'
'D1 $004013069012Z TOTAL CHG'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T2 K013 001 TOTAL VOLUMES'
'T2 A000000015005Z VOL CNT'
'T2 K015 027 TOTAL DATA SETS'
'T2 A000007044006Z DATA SET CNT'
'T3 K005 027 TOTAL'
'T3 C008025033 K/B TRK'
'T3 A004009053011Z K/B CNT'
'T4 K013 027 TOTAL CHARGES'
'T4 A004013069012$ CHRG CNT'

```

Report Record Format — (ARCHDSNVOL)

Archive / Retain Report

```

ARCHDSNVOL
*
*   Mapping of report record used to print ARCHIVE/RETAIN
*   report in DSN/VOL order
*
ARCHRECN DC      H'215,0'
          DC      CL10'ARCHDSNVOL'
NETAILIN DC      CL176' '
NOLDSNAM DC      CL44' '          OLD DATA SET NAME
NNEWNAME DC      CL44' ',CL3' '    NEW DATA SET NAME
NVOLSER  DC      CL6' '          VOLUME SERIAL
NASTRICH DC      C' '            * ASTERISK
NDEVTYPE DC      CL6' '          DEVICE TYPE
NDSORG   DC      CL3' '          DSORG
NRECFM   DC      CL2' '          RECFM
NBLKSIZE DC      CL6' '          BLKSIZE
NLRECL   DC      CL5' '          LRECL
NOPTCD   DC      CL2' '          OPTCD
NKEYLEN  DC      CL3' '          KEY LEN
NACT      DC      C' '            ACTION TAKEN
NSCRATCH DC      C' '            SCRATCHED
NCATALOG DC      C' '            CATALOGED
NCRIT1   DC      C' '            REASON
NCRIT2   DC      C' '            REASON
NCRIT3   DC      C' '            REASON
NCRIT4   DC      C' '            REASON
NKILOBYT DC      CL6' '          KILOBYTES
NTRACKS  DC      CL6' '          TRACKS
NSIMULAT DC      CL8' '          SIMULATE
NTOTARCH DC      XL2'0'          DATA SET ARCHIVED
NTOTSCR  DC      XL2'0'          DATA SET SCRATCHED
NTRKARCH DC      XL2'0'          TRACKS ARCHIVED
NTRKSCR  DC      XL2'0'          TRACKS SCRATCHED
NBYTARCH DC      XL4'0'          KILOBYTES ARCHIVED
NBYTSCR  DC      XL4'0'          KILOBYTES SCRATCHED
NTITLE   DC      C' '            REPORT TITLE

```

Table Used to Print Archive Report

```

MEMBER = ARCHDSNV
'C1 058051001132249'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 5001201032 TITLE'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K029 052 DATA SET NAME/VOLUME SEQUENCE'
'H3 K001 001132 '
'H6 K023 003 DATA SET NAME / NEWNAME'
'H6 K006 048 VOLUME'
'H6 K033 055 DEVTYPE DSORG RECFM BLKSZ LRECL '
'H6 K045 088 OPTCD KEYL ARC SCR CAT REASON KBYTES TRACKS'
'H7 K040 001 _____'
'H7 K040 041 ____ _ _ _ _ _ _ _ _ _ _ '
'H7 K040 081 ____ _ _ _ _ _ _ _ _ _ _ '
'H7 K012 121 ____ _ _ _ '
'H807008177002 SIMULATE'
'D1 C001098001 ASTERISK'
'D1 C088001003044 DSN'
'D1 C034202048006 VOLSER'
'D1 C078098055 LAST HALF OF RECORD'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K008 053 TOTAL '
'T2 K008 072 TOTAL '
'T2 K009 088 TOTAL '
'T3 K008 053 data setS'
'T3 K008 072 TRACKS '
'T3 K009 088 KILOBYTES'
'T4 K008 053 _____'
'T4 K006 073 _____'
'T4 K009 088 _____'
'T5 K021 026 ARCHIVED/BACKED UP... '
'T5 A0021850500111 TOTAL DS ARCH'
'T5 A0021890680111 TOTAL TRKS ARCH'
'T5 A0041930860111 TOTAL BYTES ARCH'
'T6 K021 026 SCRATCHED.....'
'T6 A0021870500111 TOTAL DS SCR'
'T6 A0021910680111 TOTAL TRKS SCR'
'T6 A0041970860111 TOTAL BYTES SCR'
'T7 K001 001132 '
'T8 K001 001132 '
'T9 K092 001 * AN ASTERISK BEFORE DATA SET NAME INDICATES
AN ERROR OCCURRED WHILE PROCESSING THE DATA SET'

```


Report Record Format — (ARCHVOLDSN)

Archive / Retain Report

```

ARCHVOLDSN
*
*   Mapping of report record used to print ARCHIVE/RETAIN report
*   in VOL/DSN order
ARCHRECV DC      H'215,0'
RPTNAME  DC      CL10'ARCHVOLDSN'
VETAILIN DC      CL176' '
VVOLSER  DC      CL6' '          VOLUME SERIAL
VOLDSNAM DC      CL44' '          OLD DATA SET NAME
VNEWNAME DC      CL44' ' ,CL3' '  NEW DATA SET NAME
VASTRICH DC      C' '            ASTERISK *
VDEVTYPE DC      CL6' '          DEVICE TYPE
VDSORG   DC      CL3' '          DSORG
VRECFM   DC      CL2' '          RECFM
VBLKSIZE DC      CL6' '          BLKSIZE
VLRECL   DC      CL5' '          LRECL
VOPTCD   DC      CL2' '          OPTCD
VKEYLEN  DC      CL3' '          KEY LENGTH
VACT     DC      C' '            ACTION TAKEN
VSCRATCH DC      C' '            SCRATCHED
VCATALOG DC      C' '            CATALOGED
VCRIT1   DC      C' '            REASON
VCRIT2   DC      C' '            REASON
VCRIT3   DC      C' '            REASON
VCRIT4   DC      C' '            REASON
VKILOBYT DC      CL6' '          KILOBYTES
VTRACKS  DC      CL6' '          TRACKS
VSIMULAT DC      CL8' '          SIMULATE
VTOTARCH DC      XL2'0'          DATA SET ARCHIVED
VTOTSCR  DC      XL2'0'          DATA SET SCRATCHED
VTRKARCH DC      XL2'0'          TRACKS ARCHIVED
VTRKSCR  DC      XL2'0'          TRACKS SCRATCHED
VBYTARCH DC      XL4'0'          KILOBYTES ARCHIVED
VBYTSCR  DC      XL4'0'          KILOBYTES SCRATCHED
VTITLE   DC      C' '            REPORT TITLE

```

Table Used to Print Archive / Retain Report

```

MEMBER = ARCHVOLD
'C1 058051001132249006'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 5001201032 TITLE'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K029 052 VOLUME/DATA SET NAME SEQUENCE'
'H3 K001 001132 '
'H6 K031 002 VOLUME DATA SET NAME / NEWNAME'
'H6 K032 055 DEVTYPE DSORG RECFM BLKSZ LRECL'
'H6 K045 088 OPTCD KEYL ARC SCR CAT REASON KBYTES TRACKS'
'H7 K040 001 _____'
'H7 K040 041 _____'
'H7 K040 081 _____'
'H7 K012 121 ____'
'H807008177002 SIMULATE'
'D1 C001098001 ASTERISK'
'D1 C034202002006 VOLSER'
'D1 C088007010044 DSN'
'D1 C078098055 LAST HALF OF RECORD'
'T1 K040 001 _____'
'T1 K040 041 _____'
'T1 K040 081 _____'
'T1 K012 121 _____'
'T2 K008 053 TOTAL '
'T2 K008 072 TOTAL '
'T2 K009 088 TOTAL '
'T3 K008 053 data setS'
'T3 K008 072 TRACKS '
'T3 K009 088 KILOBYTES'
'T4 K008 053 _____'
'T4 K006 073 _____'
'T4 K009 088 _____'
'T5 K021 026 ARCHIVED/BACKED UP...'
'T5 A0021850500111 TOTAL DS ARCH'
'T5 A0021890680111 TOTAL TRKS ARCH'
'T5 A0041930860111 TOTAL BYTES ARCH'
'T6 K021 026 SCRATCHED.....'
'T6 A0021870500111 TOTAL DS SCR'
'T6 A0021910680111 TOTAL TRKS SCR'
'T6 A0041970860111 TOTAL BYTES SCR'
'T7 K001 001132 '
'T8 K001 001132 '
'T9 K092 001 * AN ASTERISK BEFORE DATA SET NAME INDICATES
AN ERROR OCCURRED WHILE PROCESSING THE DATA SET'
I MIGRATE TO TAPE REPORT RECORDS

```

Report Record Format — (MIGRDSNVOL)

Migrate Report

```

MIGRDSNVOL
*
*   Mapping of report record used to print MIGRATE REPORT IN
*   DSN/VOL ORDER
MIGRREC DC      H'178',H'0'
      DC      CL10'MIGRDSNVOL'
NETAILIN DS      0CL115
NOLDSNAM DC      CL44' '
NVOLSER  DC      CL34' '
NLAHAF DS      0CL65
NDEVTYPE DC      CL6' ' ,C' '
NEXPDT   DC      CL8' ' ,CL1' '
NRECFM   DC      CL2' ' ,C' '
NBLKSIZE DC      CL6' ' ,C' '
NLRECL   DC      CL5' ' ,CL2' '
NOPTCD   DC      CL2' ' ,CL3' '
NKEYLEN  DC      CL3' ' ,CL2' '
NTAPEVOL DC      CL6' ' ,C' '
NSEQ     DC      CL3' ' ,CL2' '
NSCRATCH DC      C' ' ,CL2' '
NCATALOG DC      CL3' ' ,C' '
NREASON  DC      CL4' '
NKILOBYT DC      CL6' ' ,CL2' '
NTRACKS  DC      CL6' '
NSIMULAT DC      CL8' '
NTRKMIGR DC      XL2'0'
NBYTMIGR DC      XL4'0'
NTITLE   DC      C'6'

```

Table Used for Printing Migrate Report

```

MEMBER = MIGRDSNV
'C1 058051001132201'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 6001130032070TITLE'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K029 052 DATA SET NAME/VOLUME SEQUENCE'
'H3 K001 001132 '
'H4 K018 054 DEVICE REC'
'H4 K004 097 TAPE'
'H5 K013 001 DATA SET NAME'
'H5 K006 046 VOLUME'
'H5 K036 054 TYPE EXPDATE FM BLKSZ LRECL OPTCD'
'H5 K043 090 KEYL VOLUME SEQ SCR CAT REAS KBYTE TRACKS'
'H6 K040 001 -----'
'H6 K040 041 ---- -'
'H6 K040 081 --- - - - - -'
'H6 K012 121 ---- -'
'H707008116001 SIMULATE LINE'
'D1 C044001001044 DSN'
'D1 C034045046006 VOLSER'
'D1 C065079054065 LAST HALF OF RECORD'
'D1 F004154119006Z KILOBYTES'
'D1 H002152127006Z TRACKS'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K024 001 TOTAL DATA SETS MIGRATED'
'T2 A000 0260111 TOTAL DS MIGR'
'T2 K015 098 TOTAL KILOBYTES'
'T2 A0041541140111 TOTAL BYTES MIGR'
'T3 K012 098 TOTAL TRACKS'
'T3 A0021521220111 TOTAL TRKS MIGR'
'T6 K001 001132 '
'T7 K001 001132 '
'T8 K061 001 REASON CODES: M=MINSIZE X=MAXSIZE L=LASTUS',
'E G=GENER'
'T9 K062 001 * A PLUS SIGN AFTER TAPE VOLUME INDICATES MULTIPLE ',
'VOLUMES'

```

Report Record Format — (MIGRVOLDSN)

Migrate Report

```

MIGRVOLDSN
*
*   Mapping of report record used to print MIGRATE report in
*   VOL/DSN order
MIGRRECV DC    H'178',H'0'
          DC    CL10'MIGRVOLDSN'
VETAILIN DS    0CL115
VVOLSER  DC    CL34' '
VOLDSNAM DC    CL44' '
VLAETHAF DS    0CL65
VDEVTYPE DC    CL6' ',C' '
VEXPDT   DC    CL5' ',CL3' '
VRECFM   DC    CL2' ',C' '
VBLKSIZE DC    CL6' ',C' '
VLRECL   DC    CL5' ',CL2' '
VOPTCD   DC    CL2' ',CL3' '
VKEYLEN  DC    CL3' ',CL2' '
VTAPEVOL DC    CL6' ',C' '
VSEQ     DC    CL3' ',CL2' '
VSCRATCH DC    C' ',CL2' '
VCATALOG DC    CL3' ',C' '
VREASON  DC    CL4' '
VKILOBYT DC    CL6' ',CL2' '
VTRACKS  DC    CL6' '
VSIMULAT DC    CL8' '
VTRKMIGR DC    XL2'0'
VBYTMIGR DC    XL4'0'
VTITLE   DC    C'6'

```

Table Used for Printing Migrate Report in VOL/DSN Order

```

MEMBER = MIGRVOLD
'C1 058051001132201'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 6001130032070TITLE'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H2 K029 052 VOLUME/DATA SET NAME SEQUENCE'
'H3 K001 001132 '
'H4 K018 054 DEVICE REC'
'H4 K004 097 TAPE'
'H5 K006 001 VOLUME'
'H5 K013 008 DATA SET NAME'
'H5 K036 054 TYPE EXPDATE FM BLKSZ LRECL OPTCD'
'H5 K043 090 KEYL VOLUME SEQ SCR CAT REAS KBYTE TRACKS'
'H6 K040 001 -----'
'H6 K040 041 -----'
'H6 K040 081 -----'
'H6 K012 121 -----'
'H707008116001 SIMULATE LINE'
'D1 C034001001006 VOLSER'
'D1 C044035008044 DSN'
'D1 C065079054 LAST HALF OF RECORD'
'D1 F004154119006Z KILOBYTES'
'D1 H002152127006Z TRACKS'
'T1 K040 001 -----'
'T1 K040 041 -----'
'T1 K040 081 -----'
'T1 K012 121 -----'
'T2 K024 001 TOTAL DATA SETS MIGRATED'
'T2 A000 0260111 TOTAL DS MIGR'
'T2 K015 098 TOTAL KILOBYTES'
'T2 A0041541140111 TOTAL BYTES MIGR'
'T3 K012 098 TOTAL TRACKS'
'T3 A0021521220111 TOTAL TRKS MIGR'
'T6 K001 001132 '
'T7 K001 001132 '
'T8 K061 001 REASON CODES: M=MINSIZE X=MAXSIZE L=LASTUS',
'E G=GENER'
'T9 K062 001 * A PLUS SIGN AFTER TAPE VOLUME INDICATES MULTIPLE ',
'VOLUMES'

```

Report Record Format — (VSAMCLDATA)

VSAM Cluster Data Record

```

VSAMCLDATA
*   Mapping of report record used to print VSAM CLD Report *
VSAMREC  DC   H'2141',H'0'
          DC   CL2'***'
          DC   CL8'ADSVS026'   REPORT MODULE
C$NAME   DS   CL44             CLUSTER NAME
C$CRDT   DS   XL4             CLUSTER CREATION DATE
C$EXDT   DS   XL4             CLUSTER EXPIRATION DATE
C$VOLS   DS   CL6             PRIM. VOLSER OF DATA COMPONENT
C$VOLT   DS   XL2             PRIM. VOLSER'S DEVICE TYPE
*       NOTE: THE CLUSTER'S PASSWORDS WILL NOT BE PRESENT IN THIS
*       RECORD IF THE USER IS NOT AUTHORIZED TO SEE THEM
C$PASS   DS   0CL32           CLUSTER'S PASSWORDS
          DS   4CL8
C$TRKS   DS   F               TOTAL TRACKS ALLOC. TO CLUSTER
C$IDLE   DS   F               TOTAL NUMBER OF IDLE TRACKS
C$PCTUSE DS   H               CLUSTER PERCENT USED
C$PASATT DS   H               PASSWORD ATTEMPTS ALLOWED
C$NOEXT  DS   H               TOTAL NUMBER OF EXTENTS
*       THE FORMAT OF THE EACH EXTENT IS:
*       1. TYPE OF EXTENT      XL2   X'10' = INDEX
*       2. LOW CCHH           XL4
*       3. HIGH CCHH          XL4
*       4. TRACKS IN EXTENT    HW
*       5. VOLSER              CL6
*       6. DEVICE TYPE         XL2
C$EXTENT DS   0XL640          DATA AND INDEX EXTENT INFO.
C$EXT    DS   32XL20
C$LEXT   DS   0C              ADDRESS OF LAST EXTENT
D$ATTR   DS   XL4             DATA AND CLUSTER ATTRIB. BYTES
          ORG   D$ATTR
D$$ATTR  DS   XL2             ATTR1 AND ATTR2 OF DATA COMP.
D$ATTR3  DS   X               ATTRIBUTE FLAGS
D$ATTR4  DS   X               ATTR1 OF CLUSTER COMPONENT
D$UCAT   EQU  X'F0'           RECORD DESCRIBES A USER CATALOG
D$PAGE   EQU  X'01'           CLUSTER IS A PAGE DATA SET
D$SWAP   EQU  X'02'           CLUSTER IS A SWAP DATA SET
D$ALLOC  DS   XL7             ALLOCATION QUANTITY AND TYPE
          ORG   D$ALLOC
D$PRIM   DS   XL3             PRIMARY ALLOCATION AMOUNT
D$SEC    DS   XL3             SECONDARY ALLOCATION AMOUNT
D$TYPALO DS   X               ALLOCATION TYPE
D$CYLALO EQU  X'C0'           ALLOCATION IS IN CYLINDERS
D$TRKALO EQU  X'80'           ALLOCATION IS IN TRACKS
D$RECSZ  DS   F               AVERAGE LOGICAL RECORD SIZE

```

D\$AMD	DS	XL96	AMDSB STATISTICS
D\$HARBA	DS	F	HIGH ALLOC. RBA OF DATA COMP.
D\$HURBA	DS	F	HIGH USED RBA OF DATA COMP.
D\$TRKS	DS	F	TRACKS ALLOCATED TO DATA COMP.
D\$IDLE	DS	F	IDLE TRACKS IN DATA COMP.
D\$PCTUSE	DS	H	PERCENT USED OF DATA COMP.
I\$ATTR	DS	XL4	INDEX ATTRIBUTE BYTES
	ORG	I\$ATTR	
I\$\$ATTR	DS	XL2	ATTR1 AND ATTR2 OF INDEX COMP.
I\$ATTR3	DS	X	NOT USED
I\$ATTR4	DS	X	NOT USED
I\$ALLOC	DS	XL7	ALLOCATION QUANTITY AND TYPE
	ORG	I\$ALLOC	
I\$PRIM	DS	XL3	PRIMARY ALLOCATION AMOUNT
I\$SEC	DS	XL3	SECONDARY ALLOCATION AMOUNT
I\$TYPALO	DS	X	ALLOCATION TYPE
I\$CYLALO	EQU	X'C0'	ALLOCATION IS IN CYLINDERS
I\$TRKALO	EQU	X'80'	ALLOCATION IS IN TRACKS
I\$AMD	DS	XL96	INDEX AMDSB STATISTICS
I\$HARBA	DS	F	HIGH ALLOC. RBA OF INDEX COMP.
I\$HURBA	DS	F	HIGH USED RBA OF INDEX COMP.
I\$TRKS	DS	F	INDEX COMPONENT TRACKS
I\$IDLE	DS	F	IDLE TRACKS IN INDEX COMP.
I\$PCTUSE	DS	H	PERCENT USED OF INDEX COMP.
D\$NAME	DS	CL44	DATA COMPONENT NAME
I\$NAME	DS	CL44	INDEX COMPONENT NAME
C\$RVOL	DS	CL6	CRA VOLSER FOR CLUSTER
D\$RVOL	DS	CL6	CRA VOLSER FOR DATA COMPONENT
I\$RVOL	DS	CL6	CRA VOLSER FOR INDEX COMPONENT
C\$TYPE	DS	C	TYPE OF CLUS. (ENTYPE FR CAT.)
C\$CLUSTR	EQU	C'C'	RECORD DESCRIBES A BASE CLUSTER
C\$AIX	EQU	C'G'	RECORD DESCRIBES AN AIX
C\$SECF	DS	X	SECURITY FLAG (X'80' = RACF)
D\$SECF	DS	X	SECURITY FLAG (X'80' = RACF)
I\$SECF	DS	X	SECURITY FLAG (X'80' = RACF)
C\$CAT	DS	CL44	CAT. WHERE COMPONENT IS DEFINED
C\$FLAGS	DS	XL10	FLAGS (REFER TO DMSFLAGS MACRO)
C\$ASCNT	DS	H	NUMBER OF ASSOCIATION FIELDS
C\$ASC#	EQU	20	NUMBER OF ASSOC FIELDS
C\$ASCNAM	DS	CL(C\$ASC#*45)	ASSOC FIELDS - TYPE + NAME
D\$USDT	DS	XL3	LAST USE DATE (BINARY)
VSSELECT	DS	CL50	GROUPS SPECIFIED ON RPT SELECT

Report Record Format — (RPTDSNEX)

VBACKUP Extent Report

```

RPTDSNEX
*
*      Mapping of report record used to print the DATA SET
*      EXTENT report created by VBACKUP (VSAM portion)
*
WRTVSAM  DS      0CL130
WRTVSLT  DC      XL2'00'          LENGTH OF RECORD
WRTVSFL1 DC      XL2'00'          FILLER
WRTVSRPT DC      CL10'RPTDSNEXNT' REPORT NAME
WRTVSVOL DC      CL6'  '          VOLSER
WRTVSSDN DC      CL44'  '          DSN FOR SORT
WRTVSSDS DC      CL3'  '          DSORG FOR SORT
WRTVSSEX DC      XL8'0'          EXTENT INFO FOR SORT
WRTVSDSN DC      CL44'  '          DSN FOR PRINT
WRTVSDSO DC      CL3'  '          DSORG FOR PRINT
WRTVSEXT DC      XL8'0'          EXTENT INFO FOR PRINT
*
RPTDSNEX
*
*      Mapping of report record used to print the DATA SET
*      EXTENT report created by VBACKUP (non-VSAM portion)
*
WRTNVSM  DS      0CL130
WRTNVLT  DC      XL2'00'          LENGTH OF RECORD
WRTNVFL1 DC      XL2'00'          FILLER
WRTNVRPT DC      CL10'RPTDSNEXNT' REPORT NAME
WRTNVVOL DC      CL6'  '          VOLSER
WRTNVSDN DC      CL44'  '          DSN FOR SORT
WRTNVSDS DC      CL3'  '          DSORG FOR SORT
WRTNVSEX DC      XL8'0'          EXTENT INFO FOR SORT
WRTNVDSN DC      CL44'  '          DSN FOR PRINT
WRTNVDSO DC      CL3'  '          DSORG FOR PRINT
WRTNVEXT DC      XL8'0'          EXTENT INFO FOR PRINT
*
*

```

Table Used for Printing VBACKUP Extent Report

```

MEMBER = RPTDSNEX
'C1 058052001132130006'
'H113132 001 STANDARD HEADER - LINE 1'
'H1 K007 122 PAGE '
'H1 N004 129 PAGE NO'
'H1 K052 030 D A T A S E T E X T E N T R E P O R T
F O R'
'H1 K013 082 V O L U M E '
'H1 C006001096 VOLSER'
'H2 4132 001 STANDARD HEADER - LINE 2'
'H3 K001 001132'
'HC K013 001 DATA SET NAME'
'HC K003 048 DSO'
'HC K008 053 BEG CCHH'
'HC K008 063 END CCHH'
'HD K001 001132'
'D2 C044062003 DSN'
'D2 C003106048 DSORG'
'D2 X004109053004Z BEGINNING CCHH'
'D2 X004113063004Z ENDING CCHH'
I FILES DATA SET RECORDS

```

Subfile Record Format — (ARCHCMDS)

Deferred Archive Requests

```

ARCHCMDS DSECT
*
*   ARCHCMDS SUBFILE RECORD FORMAT (Deferred Archive Requests)
*
ARCDSNLR DS      0CL164
ARCUNAM DS       CL44      0 NEW DATA SET NAME
      DS       AL1(0)      44
ARCJOBID DS       CL8      45 JOB ID
ARCREQTM DS       XL2      53 TIME REQUESTED
ARCREQDT DS       0XL3     55 DATE REQUESTED
ARCREQY  DS       X        55 YEAR REQUESTED
ARCREQDD DS       H        56 DAY REQUESTED
ARCRUNTM DS       XL2      58 ARCHIVE TIME
ARCRUNDT DS       XL3      60 ARCHIVE DATE
ARCFLAGS DS       XL2      63 FLAGS, X'0000' PENDING
                                X'0001' COMPLETED OK
                                X'0002' FAILED
                                X'0004' ACTIVE
ARCMSGID DS       CL4      65 MESSAGE NUMBER
ARCVSLST DS       0CL30    69 VOLSERS FROM WHICH ARCHIVED
ARCVOLNO DS       5CL6     69
ARCDSNAM DS       CL44     99 DATA SET NAME
ARCEXPDT DS       XL3     143 EXPIRATION DATE FOR COMMAND
ARCRETPD DS       H       146 RETENTION PERIOD FOR COMMAND
ARCMBRCT DS       H       148 MEMBER COUNT
ARCDISP  DS       0CL3     150 DISPOSITION FROM COMMAND
ARCNO SCR DS       C       150 N=NOSCRATCH, S=SCRATCH
ARCNO CAT DS       C       151 N=NO ACTION, U=UNCAT, R=RECAT
ARC RECAT DS       C       152 E=EXPLICIT REQUEST
ARCPSWRD DS       CL8     153 PASSWORD
ARCDELDT DS       XL3     161 DATE TO DELETE FROM FILE

```

164

Subfile Record Format — (ARCHVOLS)

```

ARCHVOLS DSECT
*
*      archvols subfile record format (see archvol2 macro)
*
ARCKEY   DS    0CL6      + 0   ARCHVOL RECORD KEY (VOLSER,IF TAPE)
ARCID    DS     CL1      + 0   ARCHVOL KEY ID CHAR (LEN 1)
ARCNR    DS     CL5      + 1   ARCHVOL KEY NUMERIC (LEN 5)
ARCKEYCH DS     CL6      + 6   KEY OF NEXT ARCHVOL REC CHAINED
ARCVOLUM DS     CL6      +12   ACTUAL VOLUME OF ARCHIVE DATA SET
ARCDVTYP DS    XL4       +18   DEVICE TYPE (FROM UCB)
ARCSEQNO DS    XL1       +22   SEQUENCE NUMBER OF THIS RECORD
ARCFORMT DS    XL1       +23   ARCHVOLS REC FORMAT (NEW = X'02')
ARC#BYTE DS    XL6       +24   # BYTES ON ARCHIVE VOLUME
ARCDSNAM DS   CL44       +30   DATA SET NAME
ARCDSCNT DS     H        +74   NBR DATASETS ON TAPE
ARCBKCNT DS     F        +76   NBR BLOCKS ON VOLUMES
ARCBTCNT DS     F        +80   OLD NBR BYTES ON VOLUMES
ARCCTIME DS    XL4       +84   ARCHIVE TIME      HHMMSSSTH (UNSIGNED PKED)
ARCCDATE DS    XL3       +88   CREATION DATE   YYDDDD (DISCONT BINARY)
ARCEXPDT DS    XL3       +91   EXPIRATION DATE YYDDDD
ARCFEET  DS     H        +94   FEET USED ON TAPE
ARCCOUNT DS    AL1       +96   VOLUME COUNT (COUNT OF VOLUMES AT +0)
ARCCPYKY DS    CL6       +97   COPY ARCKEY
*
ARCFLAG  DS    XL1       +103  INDICATORS
ARCFDISA EQU   X'80'      VOLUME IS DISABLED
ARCFCOPY EQU   X'40'      VOLUME IS COPY VOLUME
ARCFPASS EQU   X'20'      VOLUME IS PASSWORD PROTECTED
ARCFERR  EQU   X'10'      CLOSE FAILED OR I/O BKSPACE FAILED
ARCFNCLO EQU   X'08'      VOLUME NOT CLOSED, ARCHVOLS INCOMPLETE
ARCFFULL EQU   X'04'      VOLUME FULL (NEXT DSN DID NOT FIT)
ARCFMERG EQU   X'02'      VOLUME CREATED BY MERGE
ARCF6250 EQU   X'01'      VOLUME WRITTEN AT 6250 BPI
*
ARCFLAG2 DS    XL1       +104
ARCDYN1  EQU   X'80'      M860 or Redwood FLAGS
ARCDYN2  EQU   X'40'
ARCDYN3  EQU   X'20'
ARCVOLBK EQU   X'10'      VOLUME LEVEL BACKUP
ARCIDRC  EQU   X'08'      COMPACTION BIT
ARCXCOPY EQU   X'04'      XCOPY HAS PROCESSED THIS ARCHVOL
ARCBYTFG EQU   X'02'      ARC#BYTE PRESENT
ARCFDEVT EQU   X'01'      ARCDDEVT PRESENT
*
          DS    CL2       +105  (RESERVED FOR FUTURE USE)
ARCRATIO DS    BL1       +107  EDRC COMPRESSION RATIO ON TAPE/CART
ARCDDEVT DS    XL2       +108  TDDC DEVICE TYPE

```

Subfile Record Format — (DASDSPCB)

DASD BILLING

```

DASDSPCB DSECT
*
*      DASDSPCB SUBFILE RECORD FORMAT (DASD BILLING)
*
DSBWORK  DS      0CL84
DSBVOLNO DS      CL6          VOLSER
DSBDSNAM DS      CL44          DATA SET NAME
DSBNOSCR DS      X            SCRATCH COUNT (obsolete)
DSBNOSCN DS      X            SCAN COUNT
DSBACCUM DS      F            ACCUMULATED AMOUNT
DSBCREDT DS      XL3          CREATION DATE
DSBSCNDT DS      XL3          LAST SCAN DATE
DSBUSEDt DS      XL3          LAST USED DATE
DSBFLAGS DS      X            X'80' - SCRATCHED
DSBUSRCD DS      CL12         USER CODE
DSBDEVCL DS      XL2          DEVICE CLASS
DSBALLOC DS      H            CURRENT ALLOCATION IN TRACKS
DSBMAXUS DS      H            LARGEST NUMBER OF TRACKS USED

```

Subfile Record Format — (DMSPARMS)

TSS Dynamic Restore

```
DMSPARMS DSECT
*
*      DMSPARMS SUBFILE RECORD FORMAT (TSS DYNAMIC RESTORE)
*
DMSRECOR DS      0CL100      0
DMSIDENT DS       CL8        0 ID FIELD OF 'TSOTAPES'
DMSLIMIT DS      PL2         8 LIMIT OF CONCURRENT DYNAMIC RESTORES
DMSCOUNT DS      PL2        10 DYNAMIC RESTORES IN PROGRESS
DMSUSRID DS     11CL8        12 USER IDS DOING DYNAMIC RESTORES
```

Subfile Record Format — (DMSPPOOLS)

For Tapes in the Tape Pools

```
DMSPPOOLS DSECT
*
*      DMSPPOOLS SUBFILE RECORD FORMAT (for tape pools)
*
*      OFFSET
POOLREC  DS      0CL90      0 DMSPPOOLS SUBFILE RECORD
POOLKEY  DS      0CL26      0 KEY TO THE SUBFILE RECORD
KEYPOOL  DS       CL08      0 NAME OF THE TAPE POOL
KEYTYPE  DS      CL01      8 S (SCRATCH) or U (USED)
KEYVOL   DS      CL06      9 'SCRATCH' or ACTUAL VOLUME IN KEY
KEYDATE  DS      CL05     15 DATE OF LAST STATUS CHANGE
KEYTIME  DS      PL04     20 TIME OF LAST STATUS CHANGE
KEYRESVD DS      CL02     24 (RESERVED FOR FUTURE USE)
POOLVOL  DS      CL06     26 VOLUME SERIAL OF TAPE IN THE POOL
POOLDATA DS     CL58     32 (RESERVED FOR FUTURE USE)
```

Subfile Record Format — (DSNINDEX)

For Archived Data Sets

```

DSNINDEX DSECT
*
*          DSNINDEX SUBFILE RECORD FORMAT (SEE DSNINDEX MACRO)
*
          DS      0D
&NAME     DS      0CL256
DSNDSNAM  DS      CL44      + 0      DATA SET NAME
DSNVOLNO  DS      CL6       +44      VOLSER FROM WHICH ARCHIVED
DSNATIME  DS      XL2       +50      ARCHIVE TIME (UNSIGNED PACKED=HHMM)
DSNMPBLK  DS      XL2       +52      MAX PHYSICAL BLK IN DATA SET (OBSOLETE)
          ORG      DSNMPBLK
DSNDSKTT  DS      XL2       +52      TT (OF TTRZ) OF DISK CNTRL RECORD
DSNARCDT  DS      XL3       +54      ARCHIVE DATE (DISCON. BIN. YDD)
DSNEXPDT  DS      0XL3      +57      EXPIRATION DATE (DISCON. BIN. YDD)
DSNEXPY   DS      XL1       +57      YEAR
DSNEXPDD  DS      XL2       +58      DAYS
*
DSNDSORG  DS      0XL2      +60      DATA SET ORGANIZATION (SAME AS F1 DSCB)
DSNDSRG1  DS      XL1       +60      FIRST BYTE OF DATA SET ORGANIZATION
DSNIS     EQU      X'80'      DSORG = IS (ISAM)
DSNPS     EQU      X'40'      DSORG = PS (PHYSICAL SEQUENTIAL)
DSNDA     EQU      X'20'      DSORG = DA (DIRECT ACCESS)
DSNUCAT   EQU      X'08'      ICF USER CATALOG
DSNDSDS   EQU      X'04'      ARCS-dumped dataset (for ADSTH001)
DSNPO     EQU      X'02'      DSORG = PO (PARTITIONED)
DSNU      EQU      X'01'      DSORG = U (UNMOVABLE)
DSNDSRG2  DS      XL1       +61      SECOND BYTE OF DATA SET ORGANIZATION
DSNAM     EQU      X'08'      DSORG = AM (VSAM)
*
DSNDEVCL  DS      X         +62      DEVICE CLASS
DSNSCALO  DS      X         +63      TYPE OF ALLOCATION
DSNPRTRK  DS      XL2       +64      PRIMARY TRACKS ALLOCATED
*          PRECEEDING SECTION OF 66 BYTES IS COMMON WITH ARC CONTROL RECORD
DSNDRBLK  DS      XL2       +66      DIRECTORY BLOCK COUNT
DSNMMBRS  DS      XL2       +68      MEMBER COUNT
DSNFILNO  DS      XL2       +70      FILE NUMBER
DSNBLKNO  DS      0XL4      +72      FIRST BLOCK (AND PRV FOR F6470'S)
DSNPRV    DS      XL1       +72      PRV FOR F6470'S
DSNRELBK  DS      XL3       +73      ALWAYS FIRST RELATIVE BLOCK NUMBER
DSNBYTES  DS      XL4       +76      BYTE COUNT
DSNBLKCO  DS      XL2       +80      OLD BLOCK COUNT FIELD
DSNARVOL  DS      CL6       +82      VOLSER ARCHIVED TO (PRE-RELEASE 7.0)
          ORG      DSNARVOL
DSNARKEY  DS      CL6       +82      KEY TO ARCHVOLS REC (POST-REL 7.0)
*

```

DSNFLAGS	DS	0XL10	+88	FLAGS
*				
DSNFLAG1	DS	XL1	+88	FLAG BYTE 1
DSNFLAGSC	EQU	X'80'		DSN SCRATCHED WHEN ARCHIVED
DSNFLAGUC	EQU	X'40'		DSN UNCATALOGED WHEN ARCHIVED
DSNFLAGRC	EQU	X'20'		DSN RECAT TO PSEUDO VOLSER WHEN ARCHIVED
DSNFLAGAR	EQU	X'10'		ARCHIVED DUE TO EXPLICIT ARCHIVE REQUEST
*				
DSNFLAGEX	EQU	X'08'		IF BIT X'10' IS OFF, ARCHIVED BY RETAIN
DSNFLAGBK	EQU	X'04'		ARCHIVED USING DASD EXCP SUPPORT
DSNFLAGUV	EQU	X'02'		ARCHIVED WITH INCREMENTAL BACKUP
DSNFLAGVV	EQU	X'01'		DA-U/V DATASET OR PS-U/V BACKED UP AS DA
DSN RESIDED ON VIRTUAL VOLUME				
*				
DSNFLAG2	DS	XL1	+89	FLAG BYTE 2
DSNFLAGEN	EQU	X'80'		HAD GENERIC PROFILE WHEN ARCHIVED
* (GENERIC, NO DISCRETE PROTECTION)				
DSNFLAGRF	EQU	X'40'		HAD DISCRETE PROFILE WHEN ARCHIVED
DSNFLAGAX	EQU	X'20'		DATA SET IS A VSAM ALTERNATE INDEX
DSNFLAGCB	EQU	X'10'		DATA SET IS A CATALOG BACKUP
DSNFLAGIO	EQU	X'08'		DATA SET BACKED UP WITH IO ERRORS
DSNFLAGPE	EQU	X'04'		DSORG IS PDSE (PARTITIONED EXTENDED)
DSNFLAGBT	EQU	X'02'		RACF BIT WAS ON WHEN ARCHIVED
DSNFLAGET	EQU	X'01'		EXTENT TRACK IMAGE BACKUP
*				
DSNFLAG3	DS	X	+90	FLAG BYTE 3
DSNBLKFG	EQU	X'80'		NEW DSNBLKCT FIELD HAS BEEN UPDATED
DSNSTRIP	EQU	X'40'		DSORG IS PSE (EXTENDED SEQUENTIAL)
DSNBSAM	EQU	X'20'		PSE ARCHIVED BY BSAM I/O
DSNMEMLA	EQU	X'10'		MEMBER LEVEL ARCHIVE
DSNBYTFG	EQU	X'08'		DSN#BYTE IS PRESENT
DSNSELMN	EQU	X'07'		SAMS:SELECT MANAGED BACKUP
DSNSELIB	EQU	X'04'		SAMS:SELECT INCR BACKUP
DSNSELDB	EQU	X'02'		SAMS:SELECT DIFF BACKUP
DSNSELFB	EQU	X'01'		SAMS:SELECT FULL BACKUP
*				
DSNFLAG4	DS	X	+91	FLAG BYTE 4
UNCAT	EQU	X'80'		DS WAS NOT CATLGED AT BACKUP TIME
DSNDSS	EQU	X'04'		ARCS-dumped dataset
DSNCC	EQU	X'02'		DS created using concurrent copy
*				
DSNFLAGN	DS	XL6	+92	FLAG BYTE 5-10
*				
	ORG	DSNFLAGS+9		ORG TO FLAG BYTE 10
DSNFLAGA	DS	X	+97	FLAG BYTE 10
DSNERRXX	EQU	X'40'		DO NOT USE! THIS FLAG WAS TURNED ON
* IN ERROR IN THE EOFREC AND WAS FIXED				
* BY CHECKING FOR BYTES 9&10=X'4040'				
* AND ZEROING LAST 7 BYTES IF EQUAL.				
DSNMSPFL	DS	XL1	+98	MSP FLAGS
*				

*-----
 * THE DSNSEXPT FLAG IS USED TO CONVERT THE DSNARCDT FIELD INTO THE
 * DSNLCOPY FIELD AND TO REFRESH THE DSNARCDT FIELD WITH ITS ORIGINAL
 * CONTENTS. THE DSNUPDAT FLAG IS NOW USED TO IDENTIFY THE DSNLCOPY
 * FIELD AS BEING UPDATED.
 *-----

DSNSMSFL DS	XL1	+99	SMS FLAGS
DSNSEXPT EQU	X'80'		ARCDT ALTERED DUE TO LAST BK COPY
DSNMCBYP EQU	X'08'		MGMT CLASS BYPASSED
DSNEXPNS EQU	X'04'		ALTERED DATE FLAG
DSNUPDAT EQU	X'02'		ARCDT ALTERED DUE TO LAST BKUP CPY

*

DSNSMSIN DS	0CL32	+100	SMS CLASSES AND GROUP INFORMATION
DSNSMSDC DS	CL8	+100	SMS DATA CLASS
DSNSMSMC DS	CL8	+108	SMS MANAGEMENT CLASS
DSNSMSSC DS	CL8	+116	SMS STORAGE CLASS
DSNSMSSG DS	CL8	+124	SMS STORAGE GROUP

*

DSNDS1FL DS	0XL12	+132	FMT1 DATE FIELDS
DSNDS1CD DS	XL3	+132	CREATION DATE
DSNDS1MD DS	XL3	+135	LAST MOD DATE
DSNDS1UD DS	XL3	+138	LAST USED DATE
DSNDS1ED DS	XL3	+141	EXPIRATION DATE
DSNBLKCT DS	XL4	+144	DSNBLKCT REPLACES OLD BLOCK COUNT
DSNRESR1 DS	XL18	+148	RESERVED FOR FUTURE
DSNDS1JN DS	XL8	+166	JOBNAME
DSNLCOPY DS	XL3	+174	LAST BACKUP CPY DETECTED DATE
DSNRATIO DS	XL1	+177	EDRC REDUCTION RATE
DSN#BYTE DS	XL6	+178	# BYTES ON ARCHIVE VOLUME
DSNRESR2 DS	XL25	+184	RESERVED FOR FUTURE

*

DSNALSTN DS	CL8	+209	USERID OF LAST UPDATE
DSNAREL# DS	CL6	+217	ARCHIVE RELEASE NUMBER
DSNRDTE DS	XL3	+223	DATE OF LAST RESTORE
DSNRTIME DS	XL2	+226	RESTORE TIME (UNSIGNED PACKED=HHMM)
DSNRTCNT DS	XL1	+228	COUNT OF RESTORES
DSNRESR3 DS	XL2	+229	RESERVED FOR FUTURE
DSNUSERF DS	XL25	+231	USER FIELD

Subfile Record Format — (MIGRECAT)

PS Migration to Tape

```
MIGRECAT DSECT
*
*   MIGRECAT SUBFILE RECORD FORMAT (PS Migration to Tape)
*   (control entries to recatalog data sets to backup tape)
*
CATREC   DS   0CL118      0
CATVOLNO DS   CL30        0 UP TO 5 PRIMARY VOLUMES
CATDSNAM DS   CL44        30 DATA SET NAME
CATDEVCL DS   XL4         74 DEVICE CLASS
CATVOLCT DS   H           78 VOLUME COUNT
CATCPYNO DS   CL30        80 UP TO 5 COPY TAPE VOLUME NUMBERS
CATNEWFS DS   H           110 FILE SEQUENCE NUMBER
CATCREDT DS   XL3         112 CREATION DATE OF MIGRECAT RECORD
CATEXPDT DS   XL3         115 EXPIRATION DATE
CATDDNAM DS   CL8         118 DDNAME FOR ORIGINAL VOLSER
CATVOLNO DS   CL6         126 ORIGINAL VOLSER
—
132
```

Subfile Record Format — (RACFENCD)

RACF Encoded Profile Names

```
RACFENCD DSECT
*
*   RACFENCD SUBFILE RECORD FORMAT (for discrete profiles)
*
          DS   0CL90      (LAST 3 CHARS OF PROFILE DSN ALWAYS BLANK)
RACDSNAM DS   CL44        + 0    DATA SET NAME
RACARCDT DS   XL3         +44    ARCHIVE DATE (YDD)
RACATIME DS   XL2         +47    ARC TIME (UNSIGNED PACKED=HHMM)
RACPFNAM DS   CL44        +49    RACF PROFILE NAME
```

Subfile Record Format — (RETCMDS)

Deferred Restore Requests

```

RETCMDS DSECT
*
*   RETCMDS SUBFILE RECORD FORMAT (DEFERRED RESTORE REQUESTS)
*
          DS      0F
RESDSNLR DS      0CL164
RESNUNAM DS      CL44          NEW DATA SET NAME
          DS      AL1(0)      FILLER X'00'
RESJOBID DS      CL8          JOB ID
RESREQTM DS      XL2          DRESTORE CMD SYSTEM TIME   XL2'HHMM'
RESREQDT DS      0XL3          DRESTORE CMD SYSTEM DATE     YDD
RESREQY  DS      X            DRESTORE CMD SYSTEM YEAR   XL1'YEAR'
RESREQDD DS      H            DRESTORE CMD SYSTEM DAY    XL2'DAY'
RESRUNTM DS      XL2          DISPOSE SYSTEM TIME        XL2'HHMM'
RESRUNDT DS      XL3          DISPOSE SYSTEM JULIAN DATE   YDD
RESFLAGS DS      XL2          FLAGS, X'0000' PENDING
                                X'0001' COMPLETED OK
                                X'0002' FAILED
                                X'0004' ACTIVE
RESMSGID DS      CL4          MESSAGE NUMBER
RESTOVOL DS      CL6          VOLUME RESTORED TO
RESDSNAM DS      CL44          DATA SET NAME
RESVOLNO DS      CL6          FROM VOLUME
RESEXPDT DS      XL3          EXPIRATION DATE FROM COMMAND
RESRETPD DS      H            RETENTION PERIOD FROM COMMAND
RESMBRCT DS      H            MEMBER COUNT
RESPOOL  DS      CL8          POOL NAME
RESTIMDT DS      0ZL9          DRESTORE CMD TIMEDATE PARM VALUE:
RESTIME  DS      ZL4          DRESTORE CMD TIME           CL4'HHMM'
RESDATE  DS      0CL5          DRESTORE CMD JULIAN DATE    YYDDD
RESDTYY  DS      CL2          DRESTORE CMD YEAR           CL2'YY'
RESDTDDD DS      CL3          DRESTORE CMD JULIAN DAY     CL3'DDD'
RESOPTNS DS      0CL4          OPTIONS FROM COMMAND
RESSCRAT DS      C            SCRATCH
RESNOCAT DS      C            NOCATALOG
RESCREAT DS      C            CREATE
RESERASE DS      X            ERASE
RESPSWRD DS      CL8          PASSWORD
RESDELDT DS      XL3          DATE TO DELETE FROM FILE

```

Subfile Record Format — (RETEXCLD)

Grace Periods for Restored Data Sets

```
RETExCLD DSECT
*
*   RETExCLD SUBFILE RECORD FORMAT (Grace Periods for Restored
*                                   Data Sets)
*
EXCLDREC DS    0CL82
EXCDSNAM DS    CL44      DATA SET NAME
EXCVOLNO DS    CL6       VOLSER NUMBER
EXCEXPDT DS    XL3       END DATE OF GRACE PERIOD
                   DS    XL29      FILLER
```

RECORD FORMATS — FILES DATA SET LOG RECORDS

FILES DATA SET LOG RECORD LAYOUT

```
RECORD 1:
                                     OFFSET
AUDFDSNM DS CL44      +0      FILES DATA SET NAME

RECORD 2:
AUDITFIL DSECT
*
*   FILES DATA SET LOG RECORD FORMAT
*
*                                   OFFSET
AUDITDAT DS XL4      + 0      DATE RECORDED (00YYDDDF)
AUDITIME DS XL4      +04      TIME RECORDED (HHMMSSHH)
AUDJOBNM DS CL8      +08      JOB NAME
AUDFUNCT DS CL8      +16      DMS FUNCTION
AUDFDATE DS XL4      +24      DMS FUNCTION START DATE
AUDFTIME DS XL4      +28      DMS FUNCTION START TIME
AUDSFFUN DS CL2      +32      SUBFILE FUNCTION
AUDSUBFL DS CL8      +34      SUBFILE
AUDRECRD DS XL0256   +42      THE UPDATED RECORD*
AUDXTRA  DS XL02      +298     RESERVED
* SEE SPECIFIC SUBFILE FOR SUBFILE RECORD DEFINITION
  If subfile function is 'SD' record may only contain
  the key for this entry.
```

Archive Tape Organization

The data sets archived during an execution of the archive support are stored as a single sequential data set with a record format of U (undefined). A file number, assigned to each data set at the time it is archived, is stored in the archive index and is used by SAMS:Disk to locate a given data set in order to restore it.

The access method used in accessing an archive tape is BSAM. Multiple buffering is employed during the reading and writing of data. The POINT macro is used to skip intervening blocks when positioning to a requested data set.

An internal algorithm is used to determine the approximate amount of data that can be carried on a single archive tape. If a data set for which an archive request is being processed will require more capacity than is estimated to remain on the current archive tape, a new archive tape will be mounted. This technique is used to confine each archived data set to a single reel of tape.

Note: Concurrent Copy is an exception to this rule. The BACKUPCC command will produce multi-volume ARCHVOLS, but only for one data set. In other words, if the current input data set will not fit on the current ARCHVOL, the BACKUPCC command will span across multiple volumes until the conclusion of the data set. At that time, the ARCHVOLS is closed and a new ARCHVOLS started. This is true with the exception of a VSAM sphere, which will span as many volumes as required.

Large data sets that exceed the capacity of a single tape volume will always start at the beginning of the first volume to which they are archived. The system will presently handle data sets that can be stored on up to 35 tape volumes. (For non-MVS users, the limit is five tape volumes.)

Archive Tape Format

The following section describes the format of data contained on the SAMS:Disk Archive Tape.

Archive tape volumes contain all information necessary to restore archived data sets. This information consists of both SAMS:Disk control information and physical blocks of the archived data. All information is stored on the archive volume as logical records.

Six different logical record types can occur in an archive file. These logical records are formatted together into physical blocks and are then written to the tape volume. More than one physical block may be needed to store all data for an archived data set.

Each physical block can contain both SAMS:Disk control records and user data blocks. Blocks are filled with as many logical records as will fit into the blocksize specified. Logical records will span physical blocks. Each partial record in the new

block has a split record prefix. For integrity purposes, logical records belonging to different archived data sets will never be stored in the same physical block.

Each physical block contains an 8-byte prefix area. This area contains control and block verification data. Its contents are:

OFFSET	LENGTH	DESCRIPTION
+0	2	Halfword length of physical block
+2	2	Halfword File Number. The file number starts at 1 and is incremented by one for every data set archived to the volume.
+4	4	Fullword Block Count. This count is the block number for this block within the archive volume.

Following the block prefix are the logical records that contain the control information and user data. Each logical record has a prefix that identifies its type and length. Following the logical record prefix is the logical record data. The first 8 bytes contain the record prefix; it will appear in all logical records. The format of logical record data differs for each logical record type. They are described later. The logical record prefix is formatted as follows:

OFFSET	LENGTH	DESCRIPTION
+0	4	Fullword length of the logical record. This length includes the length of the record prefix.
+4	1	Record Type Indicator. A 1-byte identifier is assigned to each record type. Based on this record type, SAMS:Disk can determine the contents and format of the remaining data in the record.
+5	3	TTR information

The following is a list of the valid logical record types:

Table 13-1. List of Valid Logical Record Types

Type	Description
C	SAMS:Disk Control Record. Its the first record generated for each data set archived and the first record for each data set.
1	Format-1 DSCB Record. This record always follows the control record. It contains an image of the format-1 DSCB for the archived data set.

Type	Description
V	Volume/Catalog Data Record. This record contains volume and catalog information necessary to allocate non-VSAM data sets or define VSAM clusters.
D	PDS Directory Data (PDS data sets only). One record is generated for every directory block.
R	Data Set, Data Block. One logical record is generated for every physical block contained on the original data set that was archived.
A	Alias Record. One record is generated for every PDS alias entry in the PDS archived. This record type is only generated when non-EXCP support is used.
E	Logical End of Data Set. This is the last record generated for every archived data set.
S	Split record of Data Set. This is a partial record that has been split between blocks.
P	PDS information record. This record contains information relating to PDS anomalies and directory statistics.
U	User catalog (ICF) record. This record type contains the alias pointers connecting a master catalog to an EDF-Catalog, extracted during EDF-Catalog backup.

Depending on the organization of the data set being archived and on the use of the full-track EXCP code or standard access method used to archive the data set, only certain record types will be present in the records on the archive tape. The following information indicates what records will be generated in the different cases:

Table 13-2. Record Types Generated for PS and DA

Type	Description
C	Control record
1	Format-1 DSCB
V	Volume/Catalog Data Record - allocation attributes
R	One or more physical data blocks
E	End of data set

For PO data sets using EXCP support, the following record types will be generated:

Table 13-5. Record Types Generated for PO

Type	Description
C	Control record
1	Format-1 DSCB
1	Volume/Catalog Data Record - allocation attributes
P	One or more directory statistics and anomalies block
D	One or more directory blocks
R	One or more physical data blocks. A data block with all x'EE' bytes indicates the end of data for a member.
E	End of data set

For PO (PDS and PDSE) data sets using non-EXCP support, the following record types will be generated:

Table 13-3. Record Types Generated for PDS and PDSE

Type	Description
C	Control record
1	Format-1 DSCB
V	Volume/Catalog Data Record - allocation attributes
D	Directory entry for each member
A	Entry for each alias entry
R	One or more physical data blocks. A series of D, A and R type records will be repeated.
E	End of data set

For ISAM data sets, the following record types will be generated:

Table 13-4. Record Types Generated for ISAM

Type	Description
C	Control record
1	Format-1 DSCB
V	Volume/Catalog Data Record - allocation attributes
R	Logical records from QISAM access method
E	End of data set

For AM (VSAM cluster) data sets, the following record types will be generated:

Table 13-6. Record Types Generated for VSAM

Type	Description
C	Control record
1	Format-1 DSCB (dummy)
V	Volume/Catalog Data Record - cluster attributes
R	One or more logical data records. The first is cluster catalog information used to allocate the cluster. Others are logical records returned by VSAM.
U	One or more alias entries from the master catalog (present only for backup copies of EDF-Catalogs)
E	End of data set

The following section describes the record formats for each logical record type on the archive volume.

Table 13-7. Record Format of Logical Record Types

Type	Offset	Length	Description
C — CONTROL RECORD	+0	44	Data set name
	+44	6	Volume serial
	+50	2	Archive time
	+52	2	(reserved)
	+54	3	Date archived
	+57	3	Archive expiration date
	+60	2	DSORG, from format-1 DSCB
	+62	1	DEVCL
	+63	1	DS1SCALO, from format-1 DSCB
	+64	2	Primary space quantity
	+66	2	Number of tracks used
	+68	2	Space allocated
	+70	2	Secondary space allocation
	+72	1	Cylinder overflow (F2 + 52) 1... data set in compressed fmt .xxx xxxx Trks per ISAM cyl overflow
	+73	1	CTLNOTRK (F2 + 66)

Type	Offset	Length	Description
1 — Format-1 DSCB	This is an exact copy of the format-1 DSCB found in the VTOC of the original volume.		
D — DIRECTORY RECORD	+0	256	EXCP support. Directory block from PDS. (See System Data Areas.)
D — DIRECTORY RECORD	+0	8	Non-EXCP support. Member name
	+8	4	TTRF
	+12	v	User data
A — DIRECTORY RECORD	+0	8	Non-EXCP support only. Member name
	+8	4	TTRF
	+12	v	User data
R — USER DATA BLOCKS	+0	v	User data physical blocks or logical records
E — END OF DATA SET	+0	10	' END FILE '
	+10	2	Directory count block
	+12	2	member count
	+14	1	'**'
	+15	2	DSNFLAGS - same as in DSNINDEX record
	+17	53	spaces
S — SPLIT RECORD	+0	v	remainder of logical record

Archive Tape Format (Volume-level Processing)

Volume-level backup tapes (that is, those created by the VBACKUP command) contain all information necessary to restore an entire DASD volume. This includes both SAMS:Disk control information and physical blocks of the DASD volume that was backed up. All information is stored on the volume-level backup tapes as track image records.

Table 13-8. Archive Tape Format (Volume-level)

Type		Offset	Length	Description
M — EXTENT RECORD	Created during volume-level backup processing. They contain the data set name and extent information for each data set.	+0	44	Non-VSAM. DSNAME from format-1 DSCB.
		+44	2	DSORG from format-1 DSCB
		+46	v	Data set extents from 1 to 16
M — EXTENT RECORD	Created during volume-level backup processing. They contain the data set name and extent information for each data set.	+0	44	VSAM. Cluster name from VVDS reader.
		+44	2	DSORG from format-1 DSCB
		+46	44	Catalog name from VVDS reader
		+90	44	Component name from VVDS reader
		+134	v	Data set extents from 1 to 123
I — I/O Error Record	Created during volume-level backup processing. They are written to the tape when an I/O error occurs on the DASD volume being backed up. They contain the TTR and pertinent sense data.			
		+0	v	TTR and sense data for I/O errors

Archive Tape Format for Volume-level Backup Tapes

The following information indicates what records will be generated in the volume-level backup tape format.

Table 13-9. Tape Format for Volume-level Backups

Type	Description
C	control record
1	format-4 DSCB which imitates the format-1 DSCB
M	extent records
R	one or more physical data blocks
I	I/O error record
E	end of data set

Chapter 14. PARMLIB

This section describes the user-defined members of the SAMS:Disk parameter library (parmlib). You add members to parmlib when you want to modify or tailor the way a particular SAMS:Disk function works; for example, a list of data sets that you want to exempt from processing.

SAMS:Disk also keeps much of its own control information in the parmlib data set. These members are not described here, but in the appropriate section(s) of this manual instead. *Appendix B* beginning on page [577](#) also identifies the parmlib members.

Command Summary:
VALIDATE

Program Executed:
ADSTS152

Special PARMLIB Members:
NAMEREFS

Purpose

All functions of SAMS:Disk use various control parameters during their execution. These parameters provide many different kinds of information. Some indicate how SAMS:Disk is to operate, some define the format of reports, some define data sets for which SAMS:Disk is to skip processing, and others provide user-dependent information to SAMS:Disk functions.

These parameter lists are specified as members of a partitioned data set known as the parameter library, or parmlib for short. The parmlib implementation allows SAMS:Disk to better tailor its operation to each user's requirements. The contents of a parmlib member can easily be changed to reflect the desired options. Once they are changed, the next execution of SAMS:Disk will use them.

The library is ready for use as soon as it is loaded from tape during the installation process. The supplied members are regarded as parameter lists to be used for internal SAMS:Disk system functions and should not be changed. Parameter lists to contain user information are placed in new members of the library. The documentation for each function of the system explains what user-defined members from parmlib it uses and the purpose of each.

Most SAMS:Disk functions that require user-created members will indicate specific names to be assigned when the member is created. Some functions allow this name to be overridden by parameters on the SAMS:Disk commands. This allows you to create several different parmlib members for use by the same function. Then you specify the member to be used at execution time.

See “*User Parmlib Members*” on page 547 for explanations of common SAMS:Disk parameter lists and the purpose of each. The documentation for each function of SAMS:Disk will further define the parmlib members used. More information will be found there.

Updating Member Data

Since the SAMS:Disk parameter library is a partitioned data set, it can be updated like any other PDS. SAMS:Disk does not provide special facilities to update it. Most installations have online editors to change members of PDSs. The standard Fujitsu-supplied utility for maintenance of PDS members is the IEBUPDTE utility. If you are unsure how to update this library, see your systems programmer.

Backup Considerations

Since the parameter library is a conventional PDS, it can be managed by the same means. Note, however, that if the parameter library is lost, SAMS:Disk restores cannot be done. Therefore it is advisable to occasionally take a backup copy of this data set using the JSECOPY utility. Following is sample JCL to backup the SAMS:Disk PARMLIB with the Fujitsu JSECOPY utility. A copy of this JCL is in the SAMS:Disk installation library in member name PARMBKUP.

```
//BACKUP EXEC PGM=JSECOPY
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,DSNAME=SAMS.DISK.PARMLIB
//SYSUT2 DD DISP=(,CATLG),DSNAME=SAMS.DISK.BACKUP.PARMLIB,
// UNIT=TAPE
```

Figure 14-1. JCL to Backup PARMLIB

The following is sample JCL to restore the SAMS:Disk PARMLIB with the Fujitsu JSECOPY utility:

```
//RESTORE EXEC PGM=JSECOPY
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DISP=OLD,DSNAME=SAMS.DISK.BACKUP.PARMLIB
//SYSUT2 DD DISP=(,CATLG),DSNAME=SAMS.DISK.PARMLIB,
// UNIT=DISK,SPACE=(CYL,(2,2))
//
```

Figure 14-2. JCL to Restore PARMLIB from Backup Tape

PARMLIB Specifications

These are the syntax rules for entries in members of parmlib:

1. A single entry can be from one to seventy-one characters long and can be continued if longer.
2. An entry must be enclosed in quotes if it contains embedded blanks or is continued. Any entry may be enclosed in quotes.
3. To continue an entry, add a comma after the ending quote.
4. To have quotes within an entry enclosed in quotes, use two quotes.
5. To include a comment, add an asterisk in column one. For a comment on a command line, it should follow one or more blanks after the entry.

Examples

An entry with no embedded blanks

```
1234567890
```

An entry with embedded blanks

```
'A SAMPLE ENTRY'
```

A continued entry

```
'A SAMPLE ENTRY THAT HAS BEEN CONTINUED ' ,  
'THROUGH TWO LINES'
```

A continued entry

```
'A SAMPLE ENTRY THAT HAS BEEN CONTINUED ' ,  
'THROUGH THREE ' ,  
'LINES'
```

An entry enclosed in quotes using quotes

```
'A SAMPLE ENTRY THAT HAS '' QUOTES '
```

An entry not enclosed in quotes using quotes

```
12345'67890
```

A comment

```
* THIS IS A COMMENT
```

An entry followed by a comment on the same line

```
'A SAMPLE ENTRY FOLLOWED WITH COMMENT' COMMENTED HERE
```

Several comments

```
* THIS IS A COMMENT  
* THAT HAS MORE THAN ONE LINE
```

Special Rules for NAMEREFS

NAMEREFS is a member in parmlib that allows users to specify alternate names for SAMS:Disk tables. It is intended for systems that have used previous releases of SAMS:Disk that supported parmlib members with 10-character names. This was before parmlib was converted to a partitioned data set format. The member contains the name in the old 10-character format and the new character name to be used in its place. This allows SAMS:Disk users to continue to specify old format names in the SAMS:Disk commands and still be properly accessed in the SAMS:Disk parameter library.

If this member is not used and a 10-character name is used on a SAMS:Disk command, SAMS:Disk truncates the 10-character name to 8 characters and searches parmlib for the character name.

1. Each entry in NAMEREFS has this format:

```
' OLDTABLENM NEWMEMBR '
```

where:	OLDTABLENM	the old table name - maximum of 10 characters
	NEWMEMBR	the new member name - maximum of 8 characters

Note: The entry must be enclosed in quotes because of the embedded blank between the two names.

Special Rules for POOLDEFS

The POOLDEFS member defines up to 150 pool names to be used by the archive/backup functions by associating a poolname with the name of the archive data set being created. If the data set is being created on tape, the pool name defines a tapepool that must exist within the SAMS:Disk files data set. If the data set is being created on disk (archival to disk), the pool name defines a diskpool that must exist as a member in the parmlib data set.

1. Each entry in POOLDEFS has the following format. Note that it must be enclosed in quotes because of the embedded blank between the two names.

```
'POOLNAME DATASETNAME'
```

where:	POOLNAME	a pool name — must be 8-characters in length
	DATASETNAME	a data set name/pattern, 1-to 44-characters, that is to be mapped to the specified pool

PARMLIB Syntax Validation Utility

This utility validates the syntax of the specified members or all members in the parmlib data set. A copy of the execution JCL is in the installation library in member PARMVALD.

```
//VALIDATE EXEC PGM=ADSMI002,PARM=ADSTS152,REGION=768K
//STEPLIB DD DSN=SAMS.DISK.LOAD,DISP=SHR
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DSN=SAMS.DISK.PARMLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
VALIDATE MEMBERS=
```

Figure 14-3. Sample JCL to Execute Syntax Validation

MEMBERS=

A list of from 1 to 50 member names to be validated for syntax rules.

User PARMLIB Members

Member Name	RETENTAU
Alternate Name	Specify on AUTABLE= parameter on command
Used By	RETAIN COMMAND - IMPLICIT ARCHIVAL

Description

The purpose of this member is to permit data set retention control where only data sets having one of a common set of data set name or volume prefixes are permitted to reside on a volume or class of volumes.

Each member entry must consist of a character volume serial and a 1- to 44-character data set name.

While the volume serial numbers do not have to be in any particular sequence, all entries for a given volume serial number or prefix must be contiguous in the member.

Format

```
'VVVVVVDDDD . . . '
```

Where:	VVVVVV	A 1- to 6-character volume serial number or a SAMS:Disk pattern value in positions 1 through 6.
	DDDD	A 1- to 44-character data set name or SAMS:Disk pattern name in positions 7 through 50.

Example

The first sample entry indicates that all "SYS /" data sets are authorized only for volumes beginning with the three characters "SYS". The second entry indicates that all "USERA." data sets must exist on volumes beginning with "TSS".

```
'SYS/ SYS. / '
'TSS/ USERA. / '
```

Billing Members in PARMLIB

Member Name	DSBCATEG DSBRATEC DSBUSERD DSBEXCLU
Used By	ACCUM COMMAND - DASD BILLING FUNCTION EXTEND COMMAND - DASD BILLING FUNCTION

Description

These members are used by the DASD Billing function of SAMS:Disk. Please turn to page 399 in the *User’s Guide* for a description of their use and format.

Billing User Code Definition Member

Member Name	USERCODY
Used By	ACCUM COMMAND - DASD BILLING FUNCTION EXTEND COMMAND - DASD BILLING FUNCTION

Description

This member assigns user codes to data set names or prefixes. It is used by the DASD Billing function.

Entries are used to define DASD Billing user code values to data set names. Data set names may be coded as full names or prefix names. If a data set name is encountered and no entry exists for it in the member, a value of spaces will be assigned. To allow a default value to be used instead of spaces, specify the last member entry with the default value desired and a data set name value of a single slash character.

Format

‘CCCCCCCCCCCCDDDD’

Where:	CCCCCC	A 1- to 12-character user code to be assigned to data set names matching the following data set name or pattern in positions 1 through 12.
	DDDD	A 1- to 44-character data set name or SAMS:Disk pattern name in positions 13 through 56.

Example

The following example shows three entries in a user code definition member. The first entry indicates that all data sets that begin with the characters “PROD” are assigned the user code “PRODUCTION”. All data sets beginning with “TEST” are

assigned the user code “DEVELOPMENT”. All other data sets are assigned the user code “UNKNOWN”.

```
' PRODUCTION  PROD. / '
' DEVELOPMENT TEST. / '
' UNKNOWN    / '
```

DASD Pool List

Member Name	DASDPOOL
Used By	RECOVER COMMAND - RECOVER FUNCTION RESTORE COMMAND - RESTORE FUNCTION

Description

Entries are listed in this member to define one or more pools of DASD volumes that share a common role. This member is used by the restore function to obtain an alternate volume for allocation when the data set being restored cannot be placed on the target volume. It is also used when the POOL parameter is specified on the RESTORE command.

The DASD pool list is a single member in the parmlib data set. Each entry contains the name of the volume to be associated with a DASD pool and the pool name. Multiple volumes may be associated with the same pool name by having multiple entries with the same pool name and different volume serials. Multiple DASD pools may be generated by using different pool names on groups of entries. Note that all entries comprising a pool must be grouped together.

Up to 60 different DASD pool names may be defined. Sysparm DASDVMAX specifies the total number of volumes that may be included in the defined pools (sum of volumes in all different pools). This is also the maximum number of volumes that may be specified for a single pool name. The default value is 100; for example, one pool of 100 volumes, or two of 50, etc.

Format

```
' VVVVVVPPPPPPPP '
```

Where:	VVVVVV	The 6-character volume serial of a volume in a pool. If it is fewer than six characters in length, trailing blanks must be provided and the entry enclosed in quotes in position 1 through 6.
	PPPPPPPP	The 1- to 8-character alphanumeric pool name. This name must be identical for all volume entries comprising a specific pool in positions 7 through 14.

Example

The following entries define two pools, a volume pool named RJEWORK and a volume pool named TSSDATA.

```
'VOL003RJEWORK'
'VOL004RJEWORK'
'VOL006RJEWORK'
'PACK01TSSDATA'
'PACK12TSSDATA'
'PACK15TSSDATA'
'PACK20TSSDATA'
```

Auto-Restore DSNAME Pool Member

Member Name	name specified by user in sysparm ARDSNPOL
Used By	AUTO-RESTORE DRIVEN FROM CATALOG MANAGEMENT

Description

Entries are listed in this member to define one or more data set pattern names that get associated with pools of DASD defined in the member DASDPOOL. This member is used by the auto-restore function to obtain a pool of volumes for allocation selection, rather than forcing the restore to a specific target volume.

The entries in this table relate either explicit or pattern data set names with a specific DASD pool to which to restore. All standard SAMS:Disk pattern-matching capabilities are supported for the data set names. When a data set is being restored, the entries in this table are searched from top to bottom to find a match. The first match that is found will have its pool name used. With this in mind, you should place the more specific entries at the start of the table.

If you always wanted to force a default pool to be used when no other matches are found, use the following entry as the last entry in the table:

```
'DFLTPOL/'
```

There is currently no limit to the number of entries permitted in this table. The only requirement is that all pool names referenced must exist in the member called DASDPOOL. The same pool name can be used in multiple entries in the table.

Format

' P P P P P P P P D D D D D D D D D '

Where:	PPPPPPPP	The 8-character pool name defined in member DASDPOOL in positions 1 through 8.
	DD...DD	The 1- to 44-character data set name pattern to compare against in positions 9 through 52.

Example

The following entries define several patterns to compare against. Note that the entries go from most specific to most generalized.

```
' POOLF493LABS.DMSOS.*.LOAD '
```

```
' POOF6425LABS.DMSOS.*.FILES '
```

```
' POOLWORKLABS.TEST/ '
```

```
' POOLPRODLABS.PROD/ '
```

```
' POOLWORKLABS.TJP/ '
```

```
' POOF6425LABS./ '
```

```
' POOLF493/ '
```

Auto-Restore Source Volume Pool Member

Member Name	name specified by user in sysparm ARVOLPOL
Used By	AUTO-RESTORE DRIVEN FROM CATALOG MANAGEMENT

Description

Entries are listed in this member to define one or more source volume pattern names that get associated with pools of DASD defined in the member DASDPOOL. This member is used by the auto-restore function to obtain a pool of volumes for allocation selection, rather than forcing the restore to a specific target volume.

The entries in this table relate either explicit or pattern source volume names with a specific DASD pool to which to restore. All standard SAMS:Disk pattern-matching capabilities are supported for the volume names. When a data set is being restored, the entries in this table are searched from top to bottom to find a match. The first match that is found will have its pool name used. With this in mind, you should place the more specific entries at the start of the table.

If you always wanted to force a default pool to be used when no other matches are found, use the following entry as the last entry in the table:

' DFLTPOOL/ '

There is currently no limit to the number of entries permitted in this table. The only requirement is that all pool names referenced must exist in the member called DASDPOOL. The same pool name can be used in multiple entries in the table.

Format

'PPPPPPPPVVVVVV'

Where:	PPPPPPPP	The 8-character pool name defined in member DASDPOOL in positions 1 through 8.
	VVVVVV	The 1- to 6-character pattern to compare against the source volume of the data set being restored in positions 9 through 14.

Example

The following entries define several patterns to compare against. Note that the entries go from most specific to most generalized.

'POOLF493LABS5/'
'POOF6425LABS8/'
'POOLWORKWRK/'
'POOLPRODPD/'
'POOLTEMP/'

Exclusion Table for Data Compression

Member Name	Specified in sysparm DCEXCTBL_____
Used By	ARCHIVE AND BACKUP FUNCTION

Description

Place the data set names and patterns that are to be excluded from data compression during archive or backup into this member. A match causes data compression to be bypassed.

Format

'DDDD.....'

Where:	DDDD...	A 1- to 44-character data set name or SAMS:Disk pattern data set name. Located in positions 1 through 44.
--------	---------	---

Example

The following entries cause data sets with a prefix of "SYS1" and data sets with only three nodes and the third node is "PRINT" to be excluded from compression processing.

```
'SYS1/'
'*.*.PRINT'
```

Inclusion Table for Data Compression

Member Name Specified in sysparm DCINCTBL_____

Used By ARCHIVE AND BACKUP FUNCTION

Description

Place the data set names and patterns that are to be included for data compression during archive or backup into this member. A match causes the data set to be compressed.

Format

```
'DDDD.....'
```

Where: DDDD..... A 1- to 44-character data set name or
SAMS:Disk pattern data set name.
Located in positions 1 through 44.

Example

The following entries cause data sets with a prefix of "SYS1" and data sets with only three nodes and the third node is "PRINT" to be compressed.

```
'SYS1/'
'*.*.PRINT'
```

Exemption Lists - Generalized

Member Name Specified with command parameters;
EXEMPT=
EXTABLE=
TABLE=

Used By RELEASE COMMAND (IDLE SPACE RELEASE)
SCAN COMMAND (PDS COMPRESS)
EXCLUDE COMMAND (DSCL)
MOVE/COPY COMMANDS

Description

Place the data set names and patterns that are to be exempt from SAMS:Disk volume scan processing in this member of parmlib. When SAMS:Disk is searching

DASD volumes for data sets to process, it will check to ensure that the data set name is not in this list before attempting processing for it. If the data set name is in this list, it will be exempted from processing.

Format

' DDDD '

Where: DDDD..... A 1- to 44-character data set name or
SAMS:Disk pattern data set name.
Located in positions 1 through 44.

Example

The following entries cause data sets beginning with the five characters "SYS./", and data sets with a third node level of "PRINT" to be excluded from SAMS:Disk processing.

' SYS / '
' * . * . PRINT '

Exemption List for Implicit Archive (Retain)

Member Name	RETENTXM
Alternate Name	Specify on EXTABLE= parm on command
Used By	RETAIN COMMAND - IMPLICIT ARCHIVAL

Description

Data sets that are to be exempted from retention control processing must have their names or a matching name prefix placed in this member. When EXEMPT is coded on the RETAIN command, this member is searched for a matching data set name for each data set being examined. A match causes no further processing; that is, the data set is exempt.

Member entries may be a combination of data set names and patterns with a length of 1 to 44 characters.

Note: A different member name may be used by specifying it in the EXTABLE= parameter. No exemption member is ever examined, however, unless the EXEMPT parameter is also present.

Format

' DDDD '

Where: DDDD..... A 1- to 44-character data set name or
SAMS:Disk pattern data set name. Located
in positions 1 through 44.

Example

The following entries cause data sets beginning with the five characters “SYS /” and data sets with a third node level of “PRINT” to be excluded from processing.

```
'SYS /'
'*.*.PRINT'
```

GDG DASD Generations List

Member Name	GDGDASDG (for sequential migrate function) xxxxxxxx (specified on RETGDGNM sysparm for the retain function)
Used By	SCAN COMMAND - SEQUENTIAL MIGRATE TO TAPE RETAIN COMMAND - IMPLICIT ARCHIVAL

Description

Used by the sequential migration to tape function, and the implicit archival function, this member is used to specify the number of generations of each GDG data set that are to be kept on disk.

For the sequential migrate to tape function, the older generations are copied to tape and recataloged. The entries in this member are retrieved and used only when the GENER= parameter is specified with the SCAN command. The member entries override the value specified on the GENER= parameter.

For the implicit archival function, the older generations are processed according to the options specified with the G selection type entry in the volspec table, with the member entries overriding the limit value specified in that same entry.

This data set name list is searched from top to bottom during SAMS:Disk processing. The first data set or pattern name in the list that matches the data set being processed will be used.

Format

```
'NNNNDDDD.....'
```

Where:	NNNN	A 4-digit generation count indicating the number of generations of a GDG data set to keep on disk. Located in positions 1 through 4.
	DDDD....	A 1- to 36-character GDG base name or pattern to which this version count applies. Do not include the GggggVvv as part of your pattern! Use all or a portion of the base name with an appended slash. Located in positions 5 through 40.

Example

The following sample entries indicate to keep the two current generations of all GDGs starting with PROJ.GDG.INDX1, and to keep three copies of those starting with DEPT75.GDG.INDX2. All other data sets will have one copy kept on DASD.

```
'0002PROJ.GDG.INDX1/'
'0003DEPT75.GDG.INDX2/'
'0001/'
```

Migration Exemption List

Member Name	MIGRATXM
Alternate Name	Specify on METABLE= parameter of command
Used By	SCAN COMMAND - SEQUENTIAL MIGRATE TO TAPE

Description

Place the data set names and patterns that are to be exempt from the sequential migration to tape function in this member. Any data set that matches a specification in this list will be ignored by SAMS:Disk. Each entry consists only of a 1- to 44-character data set name or pattern.

Format

```
'DDDD.....'
```

Where: DDDD..... A 1- to 44-character data set name or SAMS:Disk pattern data set name. Located in positions 1 through 44.

Example

The following entries cause data sets beginning with the five characters "SYS /" and data sets with a third node level of "PRINT" to be excluded from sequential migration processing.

```
'SYS /'
'*.*.PRINT'
```

PDS Anomaly Table

Member Name	Specify on sysparm PDSANMLY
Used By	Archive, backup, restore and recover of a PDS

Description

This member is optionally used by the SAMS:Disk archive and restore processing for a PDS. It defines specific types of anomalies found in partitioned data sets, and what options to use when they are encountered.

See the *"Processing PDSs that Contain Anomalies"* topic on page [71](#) for information about the format and use of this member.

Reblocking Member

Member Name	Specify on BLKSIZE= parm of MIGRATE command
Used By	MOVE/COPY COMMAND

Description

This member is optionally used by the Move/Copy function when the user indicates that data sets are to be reblocked as they are moved to new DASD volumes.

See the *"Reblocking"* topic on page [348](#) of the *User's Guide* for information about the format and use of this member.

Retention Period By Prefix List

Member Name	RETENTPR
Alternate Name	Specify on RPTABLE= parameter of command
Used By	RETAIN COMMAND - IMPLICIT ARCHIVAL

Description

This member is used by retention control processing. When the R selection type is specified in the volspec entry, each data set being processed is matched against the names in this member to retrieve a retention period. This retention period value is added to the data set's creation date or last used date to yield an expiration date. If this date has been reached, the data set is disposed according to the disposal actions in the volspec entry. Also see the sysparm description for *RETPREFT* on page [178](#) in this manual.

The entries are loaded into memory on the first call and remain for the duration of retention control processing. The list is searched serially until an entry is found that corresponds to the passed volume serial number and data set name.

To have a retention period applied to a given data set name or pattern regardless of volume of residence, /xxxxx should be specified as the volume prefix.

Format

'NNNNVVVVVVDDDD....'

Where:	NNNN	The 4-digit retention period to be added to the creation or last used date of any data set with the name or pattern name given in the (d-d) field. Located in positions 1 through 4.
	VVVVVV	The 1- to 6-character volume serial number of the volume on which the data set exists. This may be a SAMS:Disk pattern specification. Located in positions 6 through 10.
	DDDD....	The 1- to 44-character data set name or SAMS:Disk pattern data set name to which this entry applies. Located in positions 11 through 54.

Example

The first sample entry indicates that all "SYS/" data sets are to be kept for 365 days, no matter what volume they exist on. The second entry indicates that all data sets beginning with "TEST." will be kept for only 15 days.

```
'0365/ SYS /'
'0015TSS/ TEST./'
```

Retention Control Volume Specification Table

Member Name	RETVOLSP
Alternate Name	Specify on VSTABLE= parameter on command
Used By	RETAIN COMMAND - IMPLICIT ARCHIVAL

Description

This member is used to indicate to SAMS:Disk implicit archival what data set selection criteria are in effect and what actions are to be taken against selected data sets. See the Retention Control section for a complete explanation of this member and how it is specified.

Security List for SAMS:Disk Data Set Protection

Member Name	SECURDSN
Used By	ALL SAMS:Disk functions

Description

This member is used to define data set names to be associated with SAMS:Disk resources that are to be protected. The member is currently used for the protection of subfiles within the SAMS:Disk files data set, and SAMS:Disk command names. To activate the SAMS:Disk features to protect these resources, specify sysparms SECURFIL and SECURCMD with a value of Y.

Each entry in this member describes a resource type to be protected, the name of the resource, and the data set name and volume to be associated with the resource.

For more information on how this member is used, please turn to the topic *"Resources Protected by the Security Interface"* on page 373 in this manual.

Note: If a resource is not defined in this member and security checking has been activated for the resource type, the resource is considered not to be protected and access to the resource is allowed.

Format

```
' CRR-RRVVVVVDDDD . . . '
```

Where: '	A single quote used as a delimiter. An * asterisk indicates a comment line. Located in position 1.		
C	A single character indicator specifying the type of resource to which the entry applies. Located in position 2. The following values are valid:	C	For SAMS:Disk command name.
		F	For a subfile of the files data set.
RR-RR	A 1- to 15-character name of the resource to be protected (for example, a command or subfile name). This name can be specified with SAMS:Disk pattern matching notation. Located in positions 3 through 17.		
VVVVVV	A 1- to 6-character volume serial to be passed to the security system with the data set name. (This field is required for RACF.) Located in positions 18 through 23.		
DDDD....	A 1- to 44-character data set name known to the installation security system. Located in positions 24 through 67.		

Examples

Following are examples of entries for this member.

Example 1

The following entry protects SAMS:Disk command name ARCHIVE. Data set name “SAMS.DISK.ARCHIVE.COMMAND” will be interrogated whenever the user attempts to use the ARCHIVE command. The data set name exists on volume=SYSRES. If the user does not have OUTPUT authority for this data set name, the command will be rejected.

```
' CARCHIVE  SYSRESSAMS.DISK.ARCHIVE.COMMAND '
```

Example 2

The following entry protects SAMS:Disk subfile name DSNINDEX. Data set name “SAMS.DISK.FILE.DSNINDEX” will be interrogated whenever the user attempts to use the subfile. The data set name exists on volume=SYSRES. If the user does not have appropriate authority to process the subfile (that is, to archive a data set the user must have UPDATE authority to the DSNINDEX subfile), processing will be terminated.

```
' FDSNINDEX  SYSRESSAMS.DISK.FILE.DSNINDEX '
```

Selection Lists — Generalized

Member Name	Specified with command parameters; TABLE=
Used By	SELECT COMMAND - DSCL FIND COMMAND - DSCL

Description

Place the data set names and patterns that are to be included in processing in this member of parmlib. When SAMS:Disk is searching DASD volumes for data sets to process, it will check to ensure that the data set name is in this list before attempting to process it. If the data set name is not in this list, it will be exempted from processing.

Note: If the intention of the table is for use with either the FIND or RECOVER command, then each data set name specified must include a complete HLQ (High Level Qualifier). For a futher explanation, refer to the third example below.

Format

```
' DDDD..... '
```

Where:	DDDD.....	A 1- to 44-character data set name or SAMS:Disk pattern data set name. Located in positions 1 through 44.
--------	-----------	---

Examples

The following entries cause data sets beginning with the five characters “SYS /” and data sets with a third node level of “PRINT” to be processed by SAMS:Disk.

```
'SYS/ '
'*.*.PRINT'
'PROG.!PROD'
```

System Parameter List

Member Name	SYSPARMS
Used By	ALL SAMS:Disk FUNCTIONS

Description

The SAMS:Disk system parameter list is a user-specified list of options used by SAMS:Disk during its operation. They indicate how the user wants SAMS:Disk to operate in their environment. These parameters are explained completely in the *Sysparms* section beginning on page [103](#) in this manual.

Each entry in the SYSPARMS member specifies a separate option to SAMS:Disk. System parameters have 8-character names that identify them. Each entry contains the character name followed by the option specification. Option specification is variable depending on the definition of the option.

Format

```
'XXXXXXXXVVVV....'
```

Where:	XX..XX	System parameter name. Located in positions 1 through 8.
	VVVV....	System parameter information. Located in positions 9 through nn.

Example

The following entries define two system parameters. The first parameter indicates that the VSAM support is installed on the SAMS:Disk system. The second option indicates that default retention periods for implicit archival processing is 15 days.

```
'VSAMSUPPY'
'RETRETDP0015'
```


TSS/PFD USERID Authorization List

Member Name	TSOUSERI
Used By	ALL SAMS:Disk TSS FUNCTIONS ALL SAMS:Disk PFD FUNCTIONS

Description

This member contains the user IDs of those TSS users who are to be given the ability to archive and restore data sets of any name, not just those prefixed with their user ID. SAMS:Disk system defaults allow all users this privilege. This member is used if either sysparm TSOUSRIDY or SPFUSRIDY is present, restricting users to accessing only those data sets prefixed with their user ID.

Format

'UUUUUUU'

Where: UUUUUUU The 1- to 7-character TSS user ID of the TSS users authorized to process any data set with SAMS:Disk PFD or TSS support. Located in positions 1 through 7.

Example

The following entries indicate that users "USERA" and "PGMMGRS" are authorized to process any data sets.

'USERA'
'PGMMGRS'

User-Specified Condition Codes

Member Name	CCSET
Used By	ALL SAMS:Disk FUNCTIONS

Description

The purpose of this member is to provide the user with the capability to override or not override the condition code generated by SAMS:Disk. If the user chooses to override the SAMS:Disk-generated condition code, one of four possible alternative condition codes may be selected and assigned per message number. The first entry in this member is required. The format is as follows:

\$FLG FORMAT

Format

' \$FLGXyz '

or

' nnnnbxy '

Where:	\$FLG	\$FLG		
	X	BLANK		
	y	BATCH FLAG	B	display message under BATCH
			BLANK	do not display message under BATCH
	z	TSS FLAG	T	display message under TSS
			BLANK	do not display message under BATCH
or	nnnn	4-digit SAMS:Disk message code. Located in positions 1 through 4.		
	b	blank, located in position 5.		
	x	O	override	Use SAMS:Disk condition code. Located in position 6.
		N	no override	
	y	override condition codes (located in position 7):	G	0 (good)
			I	4 (informational message)
			R	8 (resource error)
			E	12 (error)
			C	16 (command error)

Example

The following entry indicates that the user wishes to override the SAMS:Disk-generated condition code for message 3179 with a user-specified condition code of 16.

```
' 3179 OC'
```


Chapter 15. JCL DD Statements

This section provides an explanation of all DD statements within the PROCS. All DD statements (except STEPLIB) are arranged alphabetically. This allows overriding DD statements to be used much more easily, since all you need to remember is to provide them in alphabetical order.

Table 15-1. Symbolic Parameters and Recommendations

Symbolic PARMS	Recommended Defaults	Explanations
P	SAMS.DISK.Rnn.INSTALL	Library for control statements
Q	SAMS.DISK.Rnn	Qualifier for data sets
S	*	Sysout class
W	SYSDA	Workfile default unit name

Where Rnn represents the release number that is installed. For example, Release 1.1 or 9.9 would be represented as R11 or R99, respectively.

Note: These default values are recommended, but may be different due to your local naming conventions. Please examine the actual procedures as installed and being used at your location and consult your local SAMS:Disk support personnel if you need assistance.

DD Statement Explanations

//AMSRFILE

A temporary sequential data set used by the MOVE function. Critical events are logged as they occur in VSAM data set move processing. This log is used to recover the cluster in the event SAMS:Disk abends when the data set is being moved.

//ARCHIVEC

This dd statement should be provided only when duplicate backup copies of the archive tapes are desired. It may be either omitted or dummied out if duplicate tapes are to be suppressed.

DCB attributes are automatically copied from the ARCHIVE0 dd statement. It may also be dynamically allocated based upon the values of sysparms ARCCTYPE and ARCCNAME.

//ARCHIVES

This dd statement is used by SAMS:Disk to allocate a tape drive when read access of the archives is necessary. The RESTORE and RECOVER functions and RE-BUILD access the SAMS:Disk archives for read only.

//ARCHIVE0

The output medium upon which archived data sets are to be stored is referenced by this dd statement. It may be provided in the JCL, or it may also be dynamically allocated based upon the values of sysparms ARC0TYPE and ARC0NAME. If the expiration date is not supplied by the label parameter or if its value is zero, the highest expiration date from all of the data sets to be written on it will be placed in the label as the expiration date.

//ARCHIVC1 ... ARCHIVC5

Output devices for the duplicate copies created by the MERGE function, which distributes the archived data sets based upon their expiration dates.

//ARCHIVE1 ... ARCHIVE5

Output devices for the primary copies created by the MERGE function, which distributes the archived data sets based upon their expiration dates.

//ARCHOUT

A sequential data set dynamically allocated to receive the DSCL Archive commands generated with the DASD Threshold Manager.

//COMMANDS

A temporary work file used by the RESTORE function to order the explicit RESTORE requests. This is done such that the archive tape will be mounted only once, and processed only in the forward direction.

//CMDPRINT

A transaction listing of the input command stream and any command processing messages.

//COMPWORK

A temporary work file used by the PDS Compress function. It holds the PDS member data during a SCRATCH and REALLOCATE of the data set.

//COPYFILE

A data set containing a list of SAMS:Disk archive volumes that are to be copied.

//DATEWORK

A record is written in this data set for each sequential data set to be migrated. The records are subsequently retrieved in descending expiration date order.

//DISPHOLD

A temporary data set used to suspend data set disposition until a threshold value can be verified. This allows certain SAMS:Disk functions to operate in a "fail safe" mode.

//DISPWORK

A disposition record is written in this data set for subsequent reference in determining the recatalog and scratch action after successful tape creation.

//DSNEXPDT

This file is REQUIRED (live or simulate mode) only if:

- Data sets that satisfy the selection criteria specify ARCHIVE as one of the disposal actions and
- No expiration date is provided on the //ARCHIVE0 dd statement.

The file is used to determine the highest expiration date from all the data sets being archived. This date is then assigned as the expiration date for the entire archive volume.

The file is used much like an internally generated deferred request. All disposal records for any archives to be done are temporarily stored here. The highest expiration date is found, the archive volume is opened, and the disposal records are then processed to cause the actual archiving of the data sets.

//DSNIIN

This dd statement is used as a SORTIN data set.

//DSNINDEX

This data set is used to contain the sorted ARCHIVE index records used for MERGE input.

//DSNIOUT

This dd statement is used as a SORTOUT data set.

//DSNIWK01 ... WK03

These dd statements are used as SORT work files.

//DSCBS

This dd statement is used as a SORTIN data set. It contains the format-1 DSCBs selected by the billing function.

//FILEIN

This dd statement is used as a SORTIN data set.

//FILEOUT

This dd statement is used as a SORTOUT data set.

//FILES

This dd statement allocates the SAMS:Disk files data set. It contains subfiles for the DSNINDEX, ARCHVOLS and other files.

//FILEWK01 ... WK03

These dd statements are used as SORT work files.

//FILPRINT

This dd statement defines a SYSOUT data set SAMS:Disk will use to produce a files status report.

//INDEXREC

A temporary work file used by the RECOVER function to order the DSNINDEX records. This is done such that an ARCHIVE tape will be mounted only once, and processed only in the forward direction.

//MIGRATE

Data sets are migrated to this device from DASD.

//MIGRATEC

When the COPY option has been specified to obtain duplicate tapes, this dd statement is required. The DCB attributes, including expiration date for this data set, are obtained from the MIGRATE dd statement.

//MFILES

This dd statement allocates one or more SAMS:Disk files data sets for auto restore, restore, and deferred restore processing. Each files data set allocated in the concatenation is searched to find the first DSNINDEX record for a data set to be restored. The MFILES dd statement will not be used if the FILES dd statement is allocated.

//MOVPRINT

A formatted report of data sets processed for the DSCL MOVE function.

//MSGPRINT

Execution time messages are directed to this data set.

//PARMLIB

This dd statement defines the SAMS:Disk parameter library. It contains internal system parameter lists as well as any user-defined parameters.

//PARMLIBD (TSS/PFD only)

This statement defines the SAMS:Disk parameter library. It is an alternate ddname that can only be used in an online (TSS/PFD) environment if the ddname PARM-LIB is used by another product. For more information, turn to the topic *"Step 1. Identify Files, Parmlib, and Activation Codes"* on page 10 of the *Installation Guide*.

//RELPRINT

A formatted report of data sets processed for the DSCL RELEASE function.

//REPORTS

This ddname is used to reference the data set that receives the detail records in any report generating step, and the data set containing the sorted records in the print step.

//RESTART

This data set is required by the Extend function of DASD Billing. It is used to clear the billing file and for proper recovery after an abend.

//SEQFILES

This data set contains records from a FILES unload and is used for the FILES re-load.

//SMFDATA

This ddname is used in the SMFRPT procedure to reference a sequential data set containing SMF type 14 and 15 records. These records are used to produce the "Utilization by Date and Time" report.

//STEPCHAT

The STEPCHAT dd statement should be used only to override the normal catalog search techniques.

//SYSPARMS

The SYSPARMS dd statement may be used to supply overriding sysparm values at execution time.

//SYSPRINT

A formatted listing of all actions taken during the run are directed to this data set.

//TAPEPULL

This data set contains a copy of the generated "tape pull" list and restricts SAMS:Disk to only those tapes.

//VSAM

DD statement required for VSAM cluster reference.

//VSAMKEY

DD statement used to reference a VSAM index component when it resides on a different volume from the cluster's data component.

//VSAMRVOL

DD statement used to reference a volume that contains a VSAM recoverable catalog required to successfully define a VSAM cluster.

Appendix A. DSCB Appendage Fields

Allocation information regarding individual data sets that is not available in the format-1 DSCB is maintained as an appendage to the DSCB, which is passed to all non-VSAM control processing modules.

Format-1 Fields

The fields computed for the format-1 DSCB are described below. A binary format is used for all data.

Table A-1. Format-1 DSCB Appendage Fields

Offset		Length	Description
DEC	HEX		
0	(00)	140	Data Set Control Block - DSCB
140	(8C)	8	mbbccchr of DSCB (reports only)
148	(94)	2	Tracks Allocated
150	(96)	2	Tracks Idle
152	(98)	2	PDS Directory Blocks used (a value of x'7FFF' indicates it has not been calculated yet)
154	(9A)	2	PDS Directory Blocks empty
156	(9C)	2	PDS Member count (a value of x'7FFF' indicates it has not been calculated yet)
158	(9E)	2	PDS Member average directory entry length (will be zero if not calculated yet)
160	(A0)	2	PDS Member directory capacity
166	(A6)	2	Tracks in Primary Allocation
168	(A8)	2	Extent 1 tracks
170	(AA)	2	Extent 2 tracks
172	(AC)	2	Extent 3 tracks
174	(AE)	2	Extent 4 tracks
176	(B0)	2	Extent 5 tracks

Offset		Length	Description
DEC	HEX		
178	(B2)	2	Extent 6 tracks
180	(B4)	2	Extent 7 tracks
182	(B6)	2	Extent 8 tracks
184	(B8)	2	Extent 9 tracks
186	(BA)	2	Extent 10 tracks
188	(BC)	2	Extent 11 tracks
190	(BE)	2	Extent 12 tracks
192	(C0)	2	Extent 13 tracks
194	(C2)	2	Extent 14 tracks
196	(C4)	2	Extent 15 tracks
198	(C6)	2	Extent 16 tracks
200	(C8)	2	PDS Member Directory entries available
202	(CA)	2	Primary allocation by type
204	(CC)	2	Number of Extents
206	(CE)	2	PDS - number ALIAS entries in directory
208	(D0)	2	PDS - dir entries with user TTR pointers
210	(D2)	2	PDS - dir entries with note lists
212	(D4)	1	ISAM - Number of tracks per cyl overflow
213	(D5)	1	ISAM - Number of tracks for highest index
214	(D6)	2	PDS - Max directory entry length
216	(D8)	2	PDS - Min directory entry length

Format-4 Fields

Allocation information that has been accumulated from the processing of all types of DSCBs is available as an appendage to the format-4 DSCB, which is passed as one of the parameters to the report assemble modules. The fields that are described below are carried in a binary format unless otherwise indicated.

Table A-2. Format-4 DSCB Appendage Fields

Offset		Length	Description
DEC	HEX		
0	(00)	140	Format-4 DSCB - VTOC (volser in pos 1-6)
140	(8C)	8	mbbcchhr of DSCB
148	(94)	6	Volume serial
154	(9A)	4	Unit address
158	(9E)	2	Device class
160	(A0)	1	UCBSTAT
161	(A1)	1	UCBSTAB
164	(A4)	4	Date - Julian (xxyydddf)
168	(A8)	4	Time - (xhhmmssf)
172	(AC)	2	DSCB count - blank
174	(AE)	2	DSCB count - F1
176	(B0)	2	DSCB count - F2
178	(B2)	2	DSCB count - F3
180	(B4)	2	DSCB count - F4
182	(B6)	2	DSCB count - F5
184	(B8)	2	DSCB count - F6
186	(BA)	2	DSORG count - IS
188	(BC)	2	DSORG count - PS
190	(BE)	2	DSORG count - DA
192	(C0)	2	DSORG count - PO
194	(C2)	2	DSORG count - AM
196	(C4)	2	DSORG count - other

Offset		Length	Description
DEC	HEX		
198	(C6)	4	Category count - Device total
202	(CA)	4	Category count - allocated
206	(CE)	4	Category count - free space
210	(D2)	4	Category count - idle
214	(D6)	4	Category count - lost
218	(DA)	4	Free space extent - largest
222	(DE)	4	Free space extent - second-largest
226	(E2)	4	Free space extent - third-largest
230	(E6)	2	DSCBs in VTOC
232	(E8)	2	DSCBs allocated
234	(EA)	2	Free space extent counter
236	(EC)	2	Free space extent total
238	(EE)	2	blank DSCB total
240	(F0)	2	Data set allocation - 0 tracks
242	(F2)	2	Data set allocation - 1 to 50 tracks
244	(F4)	2	Data set allocation - 51 to 125 tracks
246	(F6)	2	Data set allocation - 126 to 300 tracks
248	(F8)	2	Data set allocation - over 300 tracks

Appendix B. Contents of INSTALL Library

The following procedures will be created by executing member IEBUPDATE against the member PROCS in the PROCLIB data set on the distribution tape:

Table B-1. List of PROCS in SAMS:Disk Proclib

Member	Description
ARCHIVE	Explicit archive
BILLING	DASD billing (Select and Accumulate)
COMPRES	PDS compression
CONFIGR	Volume configurator (DASD-to-DASD Migration)
DEFVOLS	Generate "tape pull list" for deferred restores
DERASE	Batch delete of deferred archive/restore cmds
DMS	DSCL invoked function
DMSAR	Started task for auto-restores
DMSGTF	Run a GTF trace of an application
DMSLDSD	List SAMS:Disk-generated RACF profiles
DMSLRAC	List alternate RACF profile x-ref
DMSPOOL	Tape pool maintenance
DMSPROF	SAMS:Disk RACF profile maintenance utility
EXTEND	DASD billing (EXTEND Totals and Clear)
FMS	RECOVER function (DSCL-like)
IXCATLG	Recatalog data sets in the archives
IXMAINT	Index maintenance
IXUPDATE	Index update utilities
LISTD	List archive indexes
LISTREQ	List deferred archive and restore requests
MERGCOPY	Create copy of new merge archvols (post-step)
MERGE	Archive merging
MIGRATE	Migrate sequential data sets to tape

Member	Description
PDS2SEQ	Create sequential data set from PDS members
PROCS	IEBUPDTE stream containing all JCL PROCS
RACFCHK1	RACF synchronization utility
REBUILD	Rebuild index entries from tape
RECOVER	Full volume recovery
RELOAD	Reload files data set
REORG	Reorganize/resequence the files data set
REPORT	General DASD reports
RESTART	Restart after REORG shutdown
RESTORE	Explicit restore
RETAIN	Retention control (auto archival and backup)
RLSE	Release idle (over-allocated) space
SMFRPT	SMF-based report
THRSHMGR	Volume Threshold Manager Utility
UNLOAD	Unload files data set
VSAMBILL	VSAM DASD billing (select and accumulate)

The following is a list of members included with the PARMLIB data set on the distribution tape:

Table B-2. List of PARMLIB Members and Descriptions

Function	Member	Description
Message	DMSMSGES	SAMS:Disk Messages
System Parameters	PARMEDIT	Sysparm defaults/validation
Files data set	FDSAMPLE	Sample Definition for All Device Types
Reports	REPORTNM	Reportname (Control Modules for each Report)
	RPTNALLO	RPT=ALLOCS
	RPTNATTR	RPT=ATTRBS
	RPTNDIST	RPT=DISTRS
	RPTNDSIN	RPT=DSINDEX
	RPTNDSUT	RPT=DSUTIL
	RPTNADSU	RPT=ADSUTI
	RPTNFREE	RPT=FREEESP
	RPTNISAM	RPT=ISAMDS
	RPTNLAYO	RPT=LAYOUT
	RPTNMAPP	RPT=MAPPER
	RPTNMEMB	RPT=MEMBER
	RPTNMVDI	RPT=MVDICT
	RPTNAMVD	RPT=AMVDIC
	RPTNBMVD	RPT=BMVDIC
	RPTNPDST	RPT=PDSTAT
	RPTNSVDI	RPT=SVICT
	RPTNDAYW	SMF Report by day and time

Function	Member	Description
DASD Billing Report	RPTNDEV D	Summary By Device Type
	RPTNDEV Y	Summary By Device Type (if sysparm DSBALTRT"1")
	RPTNVOLD	Summary By Volume
	RPTNVOL Y	Summary By Volume (if sysparm DSBALTRT"1")
	RPTNDSND	Detail By Data Set Name
	RPTNDSNY	Detail By Data Set Name (if sysparm DSBALTRT"1")
	RPTNODSD	Detail By Dsn Owner
	RPTNOWND	Summary By Owner
Merge Report	RPTNDSNM	Data Sets Merged Forward
	RPTNVOLM	Merged Volumes Status
Deferred Disposition Report	RPTNDEFD	Report From Retain Defer
PDS Compress Report	RPTNCOMP	PDS Compress Results
Archive/Retain Sorted Report	ARCHDSNV	Data Set Name Sequence
	ARCHVOLD	Volume / Data Set Name Sequence
Sequential Migration Sorted Report	MIGRDSNV	Data Set Name Sequence
	MIGRVOLD	Volume / Data Set Name Sequence
RACF Profile Report	RPTNRFDN	Data Set Name To SAMS:Disk-saved Profile Name XREF
	RPTNRFPP	SAMS:Disk-Saved Profile Name To Data Set Name XREF

Function	Member	Description
Sample Parmlib Members	SAMPARMS	Sample entries for mbr=SYSPARMS
	SAMPCSET	Sample entries for mbr=CCSET
	SAMPDEFS	Sample entries for mbr=POOLDEFS
	SAMPPDSA	Sample entries for mbr=PDSANMLY
	SAMPREFS	Sample entries for mbr=NAMEREFS
	SAMPSPEC	Sample entries for volspec table for RETAIN
	SAMPSPFO	Sample entries for PFD options
	SAMPZDMS	Sample entries for authorized SAMS:Disk Parmlibs
PFD Functions	FMT1DICT	Format-1 dictionary (online reporting)
	FMT1FLDS	Format-1 fields (online reporting)
	FMT4DICT	Format-4 dictionary (online reporting)
	FMT4FLDS	Format-4 fields (online reporting)
	MODEL	Model commands (online reporting)
	SPFMENU	Menu options and descriptions
Source Code	ADSARnnn	Auto-restore
	ADSDMnnn	General system
	ADSDSnnn	Data security
	ADSFHnnn	DASD EXCP
	ADSMEnnn	Message editors
	ADSMInnn	Module Intercept/Transfer (not distributed)
	ADSSPnnn	PFD support
	ADSTHnnn	Tape handling
	ADSTSnnn	Table support
	ADSUTnnn	Utility
	ADSVConnn	Volume Configurator
	ADSVSnnn	VSAM

Appendix C. ENQ/DEQ/Reserve Usage

Integrity processing within SAMS:Disk is performed in two distinct ways. The first, illustrated below, covers Data Sets, Index Files, Auto-Restore, Tape, and Cartridges.

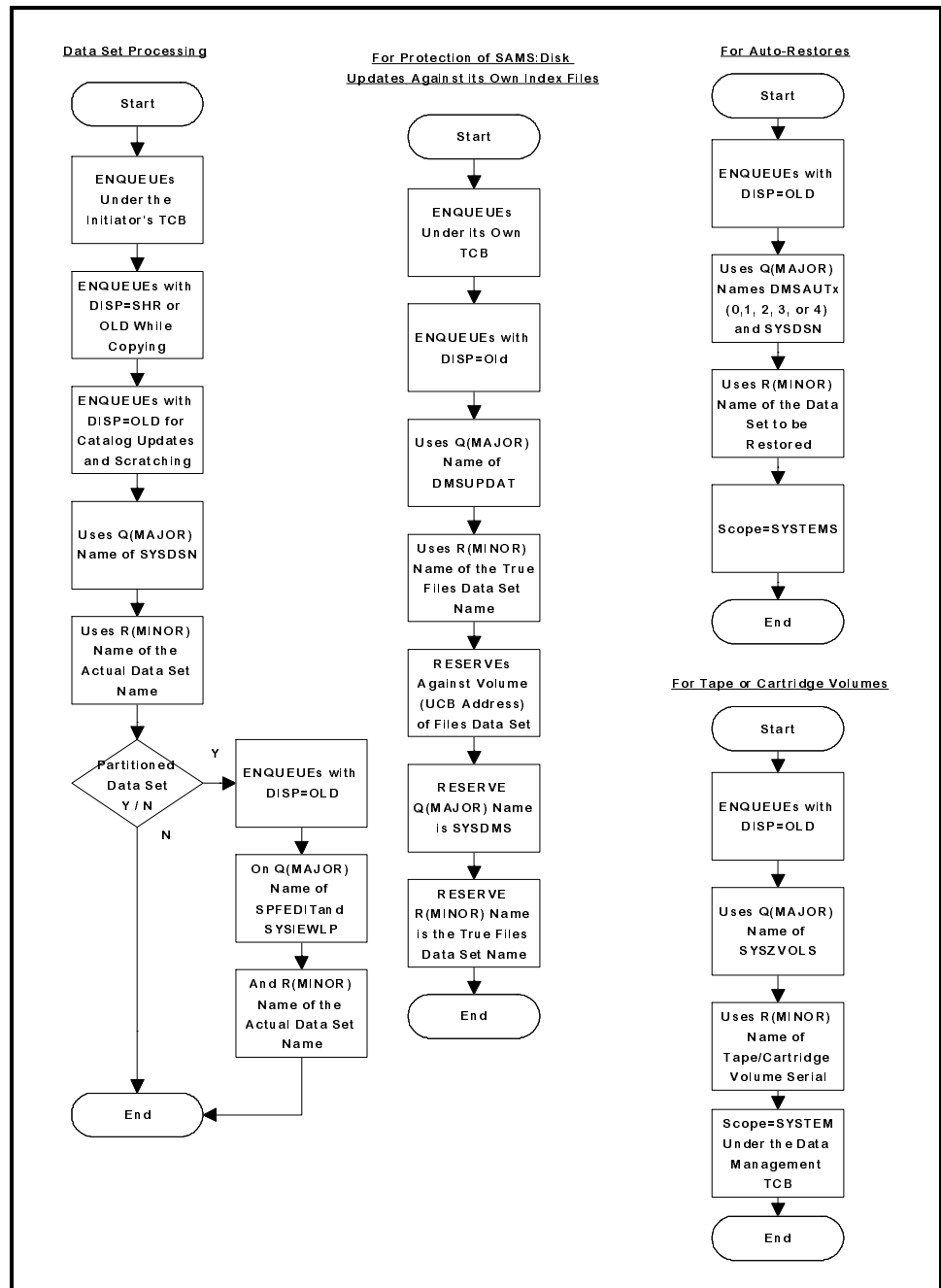


Figure C-1. Enqueue/Dequeue Reserve Usage Flowchart

The other way SAMS:Disk performs integrity is for VBACKUP and VRECOVER, which is illustrated below:

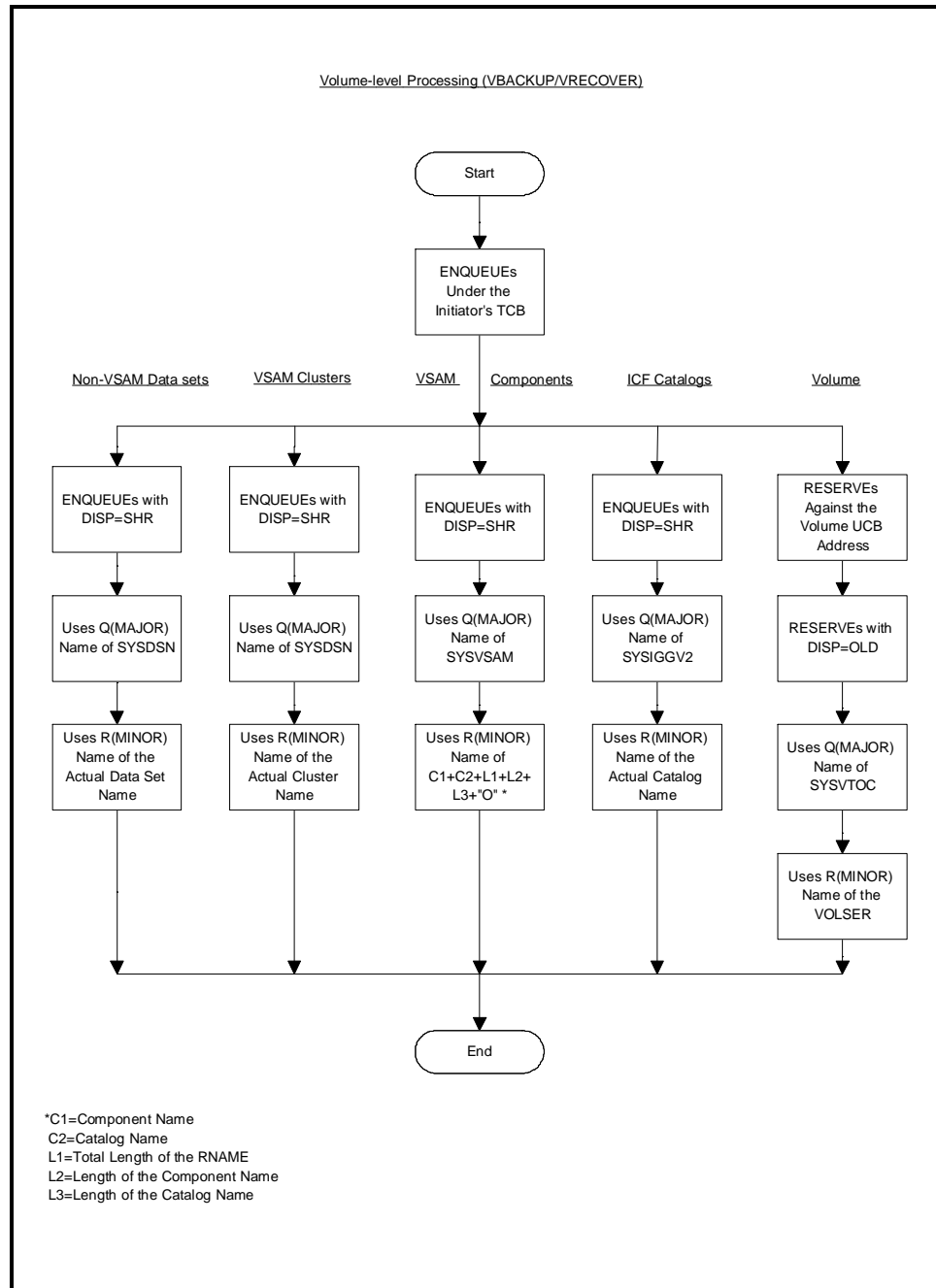


Figure C-2. Enqueue/Dequeue for VBACKUP/VRECOVER

Index

!

3420 Length Considerations	49
--------------------------------------	----

A

Abend	
Tape Status After	70
Activating	
Data Compression	58
Active Data	
Using Archive Functions to Manage	67
Adding	
Non-SAMS:Disk Modules	355
User Functions to SAMS:Disk PFD	362
Algorithms	445
Compression Results	453
Data Compression Techniques	451
Data Compression/Decompression	449
Non-VSAM Space Conversions	445
PDS Compression Space Calculations	460
PDS Compression Threshold Calculations	459
VSAM Idle Space Calculation	446
VSAM New Name, for Data and Index Component	447
Allocation	
Directory Block, PDS Compression	460
Dynamic, Versus Supplied JCL	38
of Tape Drives, Archive/Backup in Simulate	41
Secondary Space, PDS Compression	462
Space, Member Data, PDS Compression	461
Space, Specifying	65
Tape Management Considerations	38
Allocation Control Record (ACR)	3
AMSRFILE	567
Analysis	
Problem	319
Utility, Archive Tape	333
Anomalies	
PDS, Processing with	71
Anomaly	
Table, PDS	558
Appendage Fields	
Format-1 DSCB	573
Format-4 DSCB	575

ARCHCMDS	4
Archival to Disk	
Specifying	64
ARCHIVC1-5	568
Archive	
ARCHCMDS, Subfile Record Format	525
ARCHDSNVOL, Report Record Format	513
ARCHVOLDSN, Report Record Format	515
ARCHVOLS, Subfile Record Format	526
Deferred Requests, Subfile Record Format	525
DSNINDEX, Subfile Record Format	529
Extract Utility	336
Fix Utility	339
Implicit, Authorization List	548
Implicit, Exemption List for	555
Integrity and Security	69
Medium, Specifying	59
Running Multiple Jobs Concurrently	41
Specifying Blocksizes for	40
Tailoring Options	58
Tailoring Sysparms	55
Tape Analysis Utility	333
Tape Protection	69
Tape Recovery Utility	350
Tape, Concurrent Duplication	39
Archive / Retain Report	
Print Table	514, 516
Record Format	513, 515
Archive Data Set	
Assigning an Expiration Date to	62
Disk, Naming	64
Disk, Protection	71
Archive Extract Utility	336
Executing the	338
EXTRACT Command	338
JCL	338
Modifying SYSUT1 and SYSUT2	339
Output from	337
Archive Fix Utility	
Command Input	341
JCL	340
Modifying SYSUT1	340
User Exit	342
Archive Functions	
Using, to Manage Active Data	67
Archive Record Formats	
Archive Tape	535
Archive Tape, Volume-level Backup	542
Archive Tape, Volume-level Processing	541

Organization	535
Archive Tape Analysis Utility	
Command Input	334
Executing the	333
Modifying SYSUT1	334
User Exit	336
Archive Tape Recovery Utility	
Command and Parameters	350
JCL	350
ARCHIVE0	568
ARCHIVE1-5	568
ARCHIVEC	567
Archived Data Tape Format	50
Archives	568
Naming the	61
Restriction	68
Archiving to Disk	
Requirements and Recommendations	62
ARCHOUT	568
ARCHVOLS	4
Repair Utility	343
Assigning	
Expiration Date to the Archive Data Set	62
Target Volumes to Defined Pools	64
Authorization List	
for Implicit Archive	548
TSS/PFD USERID	563
Auto-Restore	22, 131, 376, 577
DSNAME Pool Member	551
Source Volume Pool Member	552

B

Backup	
Considerations, PARMLIB	544
Running Multiple Jobs Concurrently	41
Specifying Blocksizes for	40
Tailoring Options	58
Tailoring Sysparms	55
Billing	
See Also, DASD Billing	475
Block Splits	16
Blocksize	
Specifying	65
Specifying, for Archive/Backup	40

C

Calculations	
Idle Space	446
PDS Compression Threshold	459
Space, PDS Compression	460
Capacity	
Files Data Set Considerations	16
of the Files Data Set	11
Cartridge	
Switching, Length Considerations	48
Cataloging	
Tape Archives	42
CCSET	88
CCSET MEMBER	564
Checks	
Against Overwriting Valid Tapes	69
CMDPRINT	568
Coding	
for an Interactive Environment	355
PFD Report Data Conversion Modules	86
Command Input	
Archive Fix Utility	341
Archive Tape Analysis Utility	334
Commands	568
EXTRACT	338
Files Data Set	1
FRECOVER	23
Commands and Parameters	
Archive Tape Recovery Utility	350
Load Library Status Utility	346
RELOAD Utility	20
Repair Archvols Utility	344
Subfile Dump	27
Threshold Manager	95
Track Dump Utility	348
UNLOAD Utility	18
Common	
Parmlib Interface	357
Sysparms Interface	359
Compression	
and Decompression	449
Exclusion Table for	553
Inclusion Table for	554
PDS, Threshold Calculations	459
Results	453
Technique 0	452
Techniques	451
COMPWORK	568

Condition Codes	
User-specified	88, 564
Considerations	
Allocation, Tape Management	38
Archive/Backup Tailoring	58
EDRC Use and Tape Management	41
Implementation, Threshold Manager	99
Length, Cartridge Switching	48
Length, Tape Switching	48
Processing, Repair Archvols Utility	343
Tapepool	43
Control Block	
Structure, DSCL Processing	366
Control Blocks	3
Control Modules	
TSS and PFD, Relinking	355
Control Stack	
Module Intercept	324
Module Intercept, Example of	325
Module Interface, Finding the	324
Control Statement Syntax	
Threshold Manager	91
Conversions	
Space, Non-VSAM	445
COPYFILE	568
Creating	
Multivolume Tape Data Sets	40
Cross-reference	
Sysparms	105
User Exits	215
Customer Support Procedures	319
Reporting a Problem	319
Sending Information to Customer Support	323

D

DASD Billing	
DASDSPCB, Subfile Record Format	527
Device Report, Print Table	476
Device Report, Record Format	475
Members in PARMLIB	549
RPTNDEVDSB, Report Record Format	475
RPTNDSNDSB, Report Record Format	480
RPTNVOLDSB, Report Record Format	511
Subfile Record Format	527
User Code Definition Member	549
User Code Report, Print Table	481
User Code Report, Record Format	480
Volume Report, Print Table	512

Volume Report, Record Format	511
DASD Threshold Manager	90
DASDSPCB	4
Data Compression	
Activating	58
Sysparms	58
Data Conversion Modules	
PFD Report, Coding	86
Data Set Dictionary Report	
Multivolume, Print Table	473
Multivolume, Record Format	466, 472, 500
Single Volume, Print Table	509
Single Volume, Record Format	509
Data Set Index	
Subfile Record Format	529
Data Set Index Report	
Print Table	487
Report Record Format	487
Data Set Name Cross-reference Report	
Record Format	505
Data Set Names	
Map to Diskpools	64
Data Set Utilization Report	
Print Table	488
Record Format	488
DATEWORK	568
DD Statements	
AMSRFILE	567
ARCHIVC1-5	568
ARCHIVE0	568
ARCHIVE1-5	568
ARCHIVEC	567
ARCHIVES	568
ARCHOUT	568
CMDPRINT	568
COMMANDS	568
COMPWORK	568
COPYFILE	568
DATEWORK	568
Diagnostic	349
DISPHOLD	569
DISPWORK	569
DSCBS	569
DSNEXPDT	569
DSNIIN	569
DSNINDEX	569
DSNIOUT	569
DSNIWK01-03	569
Explanations	567

FILEIN	569
FILEOUT	570
FILES	570
FILEWK01-03	570
FILPRINT	570
General Information	567
INDEXREC	570
MFILES	570
MIGRATE	570
MIGRATEC	570
MOVPRINT	570
MSGPRINT	570
PARMLIB	570
PARMLIBD	571
RELPRINT	571
REPORTS	571
RESTART	571
SEQFILES	571
SMFDATA	571
SMSPRINT	571
SYSPARMS	571
SYSPRINT	571
TAPEPULL	571
VSAM	571
VSAMKEY	572
VSAMRVOL	572
Deferred	
Archive Requests, Subfile Record Format	525
Restore Requests, Subfile Record Format	533
Definition	
of Files Data Set Subfiles Entries	5
of Sysparms	123
of the Files Data Set, Subfiles Parameters	5
Dequeue Reserve Usage	583
Description	
of User Exits	221
Diagnostic Area	328
Diagnostic DD Statements	349
Diagnostic Tools	
Archive Extract Utility	336
Archive Fix Utility	339
Archive Tape Analysis Utility	333
Archive Tape Recovery Utility	350
Diagnostic DD Statements	349
Load Library Status Utility	346
Module Intercept	324
Repair Archvols Utility	343
Track Dump Utility	347
Directory Block Allocation	

PDS Compression	460
Disk Archive Data Set	
Naming	64
Protection	71
Diskpools	
Map Data Set Names to	64
DISPHOLD	569
DISPWORK	569
DMS Diagnostic Area	328
DMSAR	22, 131, 376, 577
DMSPARMS	4
DMSPOOLS	4
DSCB	
DSCBS DD Statement	569
Format-1 Appendage Fields	573
Format-2 Offsets for ISAM Report Processing	442
Format-4 Appendage Fields	575
DSCL Processing	365
Control Block Structure	366
DSNAME	
Pool Member, Auto-Restore	551
DSNAME Usage	62
DSNEXPDT	569
DSNIIN	569
DSNINDEX	569
DSNIOUT	569
DSNIWK01-03	569
Dynamic Allocation	
of Tape Units Versus Supplied JCL	38

E

EDRC	
Support, in F6470 Cartridge Devices	40
Enqueue Reserve Usage	583
Error Handling	
PDS Anomalies, Specifying Options for	72
Error Messages	
and Processing for PDS Anomalies	81
ESOTERIC	
Specifying the Unit Name	61
Estimating	
Files Data Set Capacity Requirements	11
Files Data Set Physical Space Requirements	11
Files Data Set Subfile Capacity Requirements	11
Exclusion Table for Data Compression	553
Executing	
Archive Tape Analysis Utility	333
the Archive Extract Utility	338

Exemption	
List, for Implicit Archive (Retain)	555
List, Migration	557
Lists, Generalized	554
Expiration Dates	
Assigning an, to the Archive Data Set	62
Specifying	65
Tape, Assigning	43
Extending Allocation	16
EXTRACT Command	338

F

Fast Path	
Module Intercept	326
FCB	
See Also, File Control Block	3
File Control Block	3
FILEDEFN Member	5
FILEIN	569
FILEOUT	570
FILES	570
Files Data Set	
Activating Logging	22
Allocation Control Record	3
and OVERFLOW BLOCK COUNT field	7
ARCHCMDS, Subfile Record Format	525
ARCHVOLS, Subfile Record Format	526
Block Splits	16
Capacity	11
Capacity Considerations	16
Commands	1
Control Blocks	3
DMSPARMS, Subfile Record Format	528
DMSPOOLS, Subfile Record Format	528
DSNINDEX, Subfile Record Format	529
Estimating 3380 Physical Space	12
Estimating 3390 Physical Space	14
Estimating Capacity Requirements	11
Estimating Physical Space Requirements	11
Estimating Subfile Capacity Requirements	11
Extending Allocation	16
File Control Block	3
Format of the Log Data Sets	23
Formatting of the	7
FRECOVER Command	23
FRECOVER Commands and Parameters	25
FRECOVER Internal Processing	24
FRECOVER Security Considerations	24

General Information	1
Integrity	2
Logging	22
Logging Sysparms	54
Logical Blocks	2
Logical Data Block	3
Logical Index Block	3
Logical Records	2
Management	1, 7
Managing Log Data Sets	23
Merging	29
MIGRECAT, Subfile Record Format	532
Naming Log Data Sets	22
Overflow Blocks	16
Performance Considerations of Capacity	11
Performance Tips	56
Physical Blocks	2
Programs Executed	1
RACFENCD, Subfiles Record Format	532
Recovering the	23
Recovery Report	26
RELOAD	19
RELOAD Utility	19
REORG Subfiles	21
RESTCMDS, Subfiles Record Format	533
Restrictions	5
RETEXCLD, Subfile Record Format	534
Running the FRECOVER Utility	23
Status Report	7
Status Report Field Descriptions	8
Structure	3
Subfile Dump Utility	27 - 28
Subfile Entry Definition	5
Subfiles Definition Parameters	5
Subfiles Dump Command	27
Subfiles of the	4
UNLOAD Utility	17 - 18
UNLOAD, RELOAD, and REORG Utilities	17
Utilities	17
FILEWK01-03	570
FILPRINT	570
Finding	
the Module Intercept Control Stack	324
Format	
Module Intercept Trace Control Statement	329
Format-1, -2, -3, -4 DSCB	
Format-1 Appendage Fields	573
Format-1, -2, -3, and Report Processing	437
Format-1, -4, Appendage Fields	418

Format-2 Offsets for ISAM Report Processing	442
Format-4 Appendage Fields	575
Formatting	
Message	315 - 316
Report, Tables for	419
the Files Data Set	7
FRECOVER Command	23
FRECOVER Utility	
Commands and Parameters	25
Internal Processing	24
Running the	23
Security Considerations	24

G

GDG	
DASD Generations List	556
General Information	
Files Data Set	1
Report Processing	417
Sysparms	103
Tailoring Options	51
Tape Management	37
Threshold Manager	90
User Exits	213
Generations	
List, GDG DASD	556

I

I/O Buffering	
Performance Tips	56
Idle Space	
Calculation	446
Implicit	
Archive, Authorization List	548
Archive, Exemption List for	555
Inclusion Table for Data Compression	554
INDEXREC	570
Integrity	
Archive	69
Interactive Environment	
Coding for	355
ISAM Report Processing	435
Format-2 DSCB Offsets	442
Logic Overview	438
Space Restrictions	436
Variables Examined	440

J**JCL**

Archive Extract Utility	338
Archive Fix Utility	340
Archive Tape Recovery Utility	350
Files Data Set Recovery Utility	24
Load Library Status Utility	346
Overrides	62
RELOAD Utility	19
Repair Archvols Utility	343
Subfile Dump Utility	27
Threshold Manager	95
Track Dump Utility	347
UNLOAD Utility	17
JCL DD Statements	
See also, DD Statements	567
JCL Procedures	
Members in PROCLIB	577

L**Link-edit**

and Assembly of SAMS:Disk Modules	353
---	-----

List

Authorization, for Implicit Archive	548
Authorization, TSS/PFD USERID	563
DASD Pool	550
Exemption, for Implicit Archive (Retain)	555
GDG DASD Generations	556
Migration Exemption	557
Security, for SAMS:Disk Data Set Protection	560
System Parameter	562

Lists

Exemption, Generalized	554
Selection, Generalized	561

Load Library Status Utility	346
---------------------------------------	-----

Command and Parameters	346
----------------------------------	-----

JCL	346
---------------	-----

Logging

Activating	22
Files Data Set	22
Format of the Log Data Sets	23
Managing Log Data Sets	23
Naming Log Data Sets	22

Logical Data Block (LDB)	3
------------------------------------	---

Logical Index Block (LIB)	3
-------------------------------------	---

M

Manage Active Data Using Archive Functions	67
Management	
Files Data Set	1, 7
Tape	37
Map	
Data Set Names to Diskpools	64
Member	
Data, Space Allocation	461
Data, Updating PARMLIB	544
DSNAME Pool, Auto-Restore	551
Parmedit	360
REPORTNM	417
Members	547
ARCHDSNV	514
ARCHVOLD	516
MIGRDSNV	518
MIGRVOLD	520
RPTDSNEX	524
RPTNBMVD	473
RPTNDAYW	490
RPTNDEVVD	476
RPTNDSIN	487
RPTNDSND	481
RPTNDSUT	488
RPTNMEMB	499
RPTNMVDI	500
RPTNPdst	502
RPTNRFDN	506
RPTNRFPP	508
RPTNSVDI	509
RPTNVOLD	512
Memory Requirements	
Performance Tips	56
Merging	
Files Data Sets	29
Message Formatting	315 - 316
MFILES	570
MI	
See Also, Module Intercept	324
MI Fast Path	
See Also, Module Intercept	326
Migrate	570
MIGRDSNVOL, Report Record Format	517
MIGRVOLDSN, Report Record Format	519
MIGRATEC	570
Migration	
Exemption List	557

MIGRECAT	4
Modifying	
SAMS:Disk	353
SAMS:Disk PFD Functions	364
SYSUT1 and 2 for the Archive Extract Utility	339
SYSUT1 for the Archive Fix Utility	340
SYSUT1 for the Archive Tape Analysis Utility	334
Modifying SAMS:Disk	353
Adding Non-SAMS:Disk Modules	355
Adding User Functions to SAMS:Disk PFD	362
Assembly and Link-edit of SAMS:Disk Modules	353
Coding for an Interactive Environment	355
Issuing Messages	355
Modifying SAMS:Disk PFD Functions	364
Parameter Library Support	357
Relinking TSS and PFD Control Modules	355
Sysparms Library Support	359
System Parameter Validation	360
Module Call Interface	356
Example	356
User Exits	356
Module Intercept	
Control Stack	324
Control Stack, Example of	325
Control Stack, Finding the	324
DMS Diagnostic Area	328
Fast Path	326
Trace Output	332
Trace, Control Statement Format	329
Trace, Options Control Statement	331
Tracing	328
Modules	
Control, Relinking TSS and PFD	355
Non-SAMS:Disk, Adding	355
SAMS:Disk, Assembly and Link-edit of	353
MOVPRINT	570
MSGPRINT	570
Multivolume	
Tape Data Sets, Creating	40
Multivolume Data Set Dictionary Report	472, 500
MVD	
See Also, Multivolume Data Set Dictionary	473
 N	
NAMEREFS, Special Rules	546
Naming	
the Archives	61
the Disk Archive Data Set	64

New Name Algorithm	
for VSAM Data and Index Component	447
Non-VSAM	
Space Conversions	445

O

Output	
from the Archive Extract Utility	337
Module Intercept Trace	332
Repair Archvols Utility	343
OVERFLOW BLOCK COUNT field	7
Overflow Blocks	16
Overriding	
Sysparms Instream	104

P

Parameter Library Support	357
Parameters	
System List, PARMLIB	562
Parmedit Member	360
PARMLIB	543, 570
Backup Considerations	544
Billing Members in PARMLIB	549
DASD Billing User Code Definition Member	549
DASD Pool List	550
Data Set, Purpose of	543
DESCRIPTION Member	548
Special Rules for NAMEREFS	546
Special Rules for POOLDEFS	547
Specifications	545
Syntax Rules for	545
Syntax Validation Utility	547
Updating Member Data	544
User Members	548
Parmlib Interface	357
PARMLIBD	571
PDS	
Anomaly Table	558
PDS Anomalies	
Code Descriptions and Notes	74
Conditions Not Regarded as Errors	81
Error Messages and Processing	81
Processing Defined	74
Processing with	71
Specifying Options for Error Handling	72
PDS Compress	

Performance Tips	56
Space Calculations	460
Threshold Calculations	459
PDS Compression Space Calculations	
Directory Block Allocations	460
Member Data Space Allocation	461
Secondary Space Allocation	462
PDS Compression Threshold Calculations	
Member Data Used Space Percentage	459
Used Space Percentage	459
PDS Directory (CCHHR, DUMP, TTRD)	504
PDS Directory / Member Report	
Print Table	499
Record Format	498
PDS Status Report	
Print Table	502
Record Format	502
Performance Considerations	
Files Data Set Subfiles and Capacity	11
Performance Tips	
Files Data Set	56
I/O Buffering and Memory Requirements	56
PDS Compress	56
Report Processing	56
Tailoring Options	56
Period	
Retention, by Prefix List	558
PFD Online Reports	
Tailoring	82
Pool	
Defined, Assigning Target Volumes to	64
Diskpools, Map Data Set Names to	64
List, DASD	550
Pool Member, DSNAME, Auto-Restore	551
POOLDEFS syntax	63
POOLDEFS, Special Rules	547
Prefix List, Retention Period by	558
Problem	
Reporting	319
Problem Analysis	
Diagnostic Tools	324
Problem Analysis / Aids	319
Processing	
and Error Messages for PDS Anomalies	81
Considerations, Repair Archvols Utility	343
DSCL	365
DSCL Control Block Structure	366
for Each PDS Anomaly, Defined	74
PDSs that Contain Anomalies	71

Report	417
PROCLIB Members	577
Programs Executed	
Files Data Set	1
PS Migration to Tape	
Subfile Record Format	532
Purpose of the PARMLIB Data Set	543

R

RACF	
Encoded Profile Names	532
Profile Cross-reference Print Table	508
Profile Cross-reference Report, Print Table	506
Profile Cross-reference Report, Record Format	505, 507
RACFENCD	4
Reblocking Member	558
Recommendations	
Archiving to Disk	62
Record Format, Archive	
See Also, Archive Record Formats	535
Record Format, Reports	
See Also, Report Record Formats	463
Record Formats, Subfile	
See Also, Subfile Record Formats	525
Recover	
Tailoring Sysparms	55
Recovery	
Archive Tape, Utility	350
of the Files Data Set	23
Relinking	
TSS and PFD Control Modules	355
RELOAD Utility	17
Command and Parameters	20
RELPRINT	571
REORG Utility	16 - 17, 21, 29
Repair Archvols Utility	343
Command and Parameters	344
JCL	343
Output	343
Processing Considerations	343
Report Processing	417
Detail Entry Examples	424
Format Table Examples	423
Format-1, -4, DSCB Appendage Fields	418
General Information	417
Header Entry Examples	423
ISAM	435
ISAM Format-2 DSCB Offsets	442

ISAM Space Restrictions	436
Owner Table Example	427
Record Format for Owner Record Example	429
Report Logic Diagram	432
Report Modification Example	426
REPORTNAME Table	417
Standard Multivolume Dictionary Record Format	430
Standard Multivolume Dictionary Report Table	428
Tables for Report Formatting	419
Total Entry Examples	425
VTOC Entries for Format 1, 2, and 3 DSCBs	437
Report Record Formats	
ARCHDSNVOL	513
Archive / Retain Print Table	514, 516
Archive / Retain Report	513, 515
ARCHVOLDN	515
Billing Device Report	475
DASD Billing by User Code Print Table	481
DASD Billing by User Code Report	480
DASD Billing Device Print Table	476
DASD Billing Volume Print Table	512
DASD Billing Volume Report	511
Data Set Index Print Table	487
Data Set Index Report	487
Data Set Name Cross-reference Report	505
Data Set Utilization Print Table	488
Data Set Utilization Report	488
Migrate Print Table	518
Migrate Report	517, 519
MIGRDSNVOL	517
MIGRVOLDN	519
Multivolume Data Set Dictionary Print Table	473, 500
Multivolume Data Set Dictionary Report	466, 472, 500
PDS Directory / Member Print Table	499
PDS Directory / Member Report	498
PDS Directory Report	504
PDS Status Print Table	502
PDS Status Report	502
RACF Profile Cross-reference Print Table	506, 508
RACF Profile Cross-reference Report	505, 507
RPTDSNEX	523
RPTNALLOCS	463
RPTNAMVDIC	466
RPTNATTRBS	469
RPTNBMVDIC	472
RPTNDAYWEK	490
RPTNDEVDSB	475
RPTNDISTR	477
RPTNDSINDX	487

RPTNDSNDSB	480
RPTNDSUTIL	488
RPTNFREESP	492
RPTNLAYOUT	494
RPTNMAPPER	496
RPTNMEMBER	498
RPTNMVDICT	500
RPTNPDSTAT	502
RPTNPOCCHH	504
RPTNPODUMP	504
RPTNPOTTRD	504
RPTNRACFPF	507
RPTNVOLDSB	511
Single Volume Data Set Dictionary Print Table	509
Single Volume Data Set Dictionary Report	509
Utilization by Date and Time Print Table	490
Utilization by Date and Time Report	490
VBACKUP Extent Print Table	524
VBACKUP Extent Report	523
Volume Allocation Summary Print Table	464
Volume Allocation Summary Report	463
Volume Attribute Summary Print Table	470
Volume Attribute Summary Report	469
Volume Distribution Summary Print Table	478
Volume Distribution Summary Report	477
Volume Freespace Print Table	493
Volume Freespace Report	492
Volume Layout Print Table	495
Volume Layout Report	494
Volume Mapping Print Table	497
Volume Mapping Report	496
VSAM Cluster Data Record Report	521
VSAMCLDATA	521
Reporting	
a Problem	319
Reports	571
Files Data Set Recovery	26
Files Data Set Status	7
Formatting, Tables for	419
Performance Tips	56
PFD Online, Tailoring	82
PFD, Coding Data Conversion Modules	86
Processing	417
Tailoring Sysparm for Separator Pages	54
Tailoring Sysparms for Lines Per Page	54
Tailoring User Exit for Sysout Print Lines	54
Requirements	
Archiving to Disk	62
Memory, Performance Tips	56

RESTART	571
RESTARTCMDS	5
Restore	
Deferred Requests, Subfile Record Format	533
Dynamic, Subfile Record Format	528
RETEXCLD, Subfile Record Format	534
Tailoring Sysparms	55
Restrictions	
Archiving the Archives	68
Files Data Set	5
Space, ISAM Report Processing	436
Retain	
Exclusion, Subfile Record Format	534
Retain, Exemption List for Implicit Archive	555
Retention Control	
Volume Specification Table	559
Retention Period by Prefix List	558
RETEXCLD	5
Rules	
Special, for NAMEREFS	546
Special, for POOLDEFS	547
Syntax, for PARMLIB	545
Running	
Archive/Backup in Simulate Mode	41
Concurrent Archive/Backup Jobs	41
S	
SAMPCSET	88
Secondary Space	
Allocation, PDS Compression	462
Security	
Archive	69
List for SAMS:Disk Data Set Protection	560
Selection	
Lists, Generalized	561
Sending Information to Customer Support	323
SEQFILES	571
Simulate Mode	
Running Archive/Backup	41
Single Volume Data Set Dictionary Report	
Print Table	509
Record Format	509
SMFDATA	571
Source Volume Pool Member, Auto-Restore	552
Space Allocation	
Member Data, PDS Compression	461
Specifying	65
Space Conversion	

Non-VSAM	445
Specifications	
PARMLIB	545
Specifying	
Archival to Disk	64
Blocksize	65
Expiration Date	65
Options for PDS Anomaly Error Handling	72
Space Allocation	65
the Archive Medium	59
the Media Type	60
Status	
Tape, After Abends	70
STEPCAT	571
Structure	
Control Block, DSCL Processing	366
Files Data Set	3
Subfile Dump Utility	27 - 28
Subfile Record Formats	
ARCHCMDS	525
ARCHVOLS	526
DASD Billing	527
DASDSPCB	527
Deferred Archive Requests	525
Deferred Restore Requests	533
DMSPARMS	528
DMSPOLS	528
DSNINDEX	529
For Archived Data Sets	529
Grace Periods for Restored Data Sets	534
MIGRECAT	532
PS Migration to Tape	532
RACF Encoded Profile Names	532
RACFENCDD	532
RETCMDS	533
RETEXCLD	534
Tape, in Tape Pools	528
TSO Dynamic Restore	528
Subfiles Entries	
of the Files Data Set, Definition of	5
Support	
Customer, Procedures	319
EDRC, in F6470 Cartridge Devices	40
Parameter Library	357
Sysparms Library	359
Tape GDG	42
SVD	
See Also, Single Volume Data Set Dictionary	509
Switching	

Cartridge, Length Considerations	48
Tape, Length Considerations	48
Syntax	
POOLDEFS	63
Rules for PARMLIB	545
Threshold Manager Control Statements	91
Validation Utility for PARMLIB	547
Sysparms	103, 571
DISKPROT	71
Cross-reference	105
DASDPOOL	550
Definitions	123
FLYCOUNT	54
General Information	103
Names Reserved for Users	104
Overriding, Instream	104
RETEXCLD	5
RPTLINES	54
Tailoring for Archive/Backup	55
Tailoring for Files Data Set Logging	54
Tailoring for Messages	53
Tailoring for Separator Pages	54
Tailoring Lines Per Page	54
Tailoring Restore/Recover	55
Sysparms Interface	359
Sysparms Library Support	359
SYSPRINT	571
System Parameter List	562
System Parameter Validation	360

T

Table	
Exclusion, for Data Compression	553
Inclusion, for Data Compression	554
PDS Anomaly	558
Volume Specification, Retention Control	559
Tailoring Options	51
Activating Data Compression	58
Anomalous Conditions Not Regarded as Errors	81
Archive Data Set, Assigning an Expire Date	62
Archive Disk Protection	71
Archive Integrity and Security	69
Archive Tape Protection	69
Archive/Backup Considerations	58
Archiving the Archives is Not Allowed	68
Archiving to Disk	62
Checks Against Overwriting Valid Tapes	69
Coding PFD Report Conversion Modules	86

DASD Threshold Manager	90
Data Compression Sysparms	58
DSNAME Usage	62
Error Messages and Processing for Anomalies	81
Files Data Set Performance	56
General Information	51
I/O Buffering and Memory Requirements	56
IXMAINT, MERGE, REBUILD	66
JCL Overrides	62
Map Data Set Names to Diskpools	64
Naming the Archives	61
Naming the Disk Archive Data Set	64
Other Methods	51
PDS Anomaly Code Descriptions	74
PDS Compress Performance	56
Performance Tips	56
Place Target Volumes in Defined Pools	64
Processing Defined for Each Anomaly	74
Processing PDSs that Contain Anomalies	71
Report Processing Performance	56
Specifying an Expiration Date	65
Specifying Archival to Disk	64
Specifying Blocksize	65
Specifying Options for PDS Error Handling	72
Specifying Space Allocation	65
Specifying the Archive Medium	59
Suggested System Parameters	53
System Parameter for Files Data Set Logging	54
System Parameter for Lines Per Page	54
System Parameter for Separator Pages	54
System Parameters	51
System Parameters for Archive/Backup	55
System Parameters for Messages	53
System Parameters for Restore/Recover	55
Tailoring PFD Online Reports	82
Tape Status After Abends	70
Threshold Manager Control Statement Syntax	91
User Exit for All Sysout Print Lines	54
User Exits	51
User Specified Condition Codes	88
Using Archive Functions to Manage Active Data	67
Using Multiple Types of Media	60
TAPCTLEX User Exit	289
Tape	
Archive, Concurrent Duplication	39
Archive, Recovery Utility	350
Expiration Dates, Assigning	43
Format of Archived Data	50
GDG Support	42

Management	37
Protection, Archive	69
Status After Abends	70
Switching, Length Considerations	48
Valid, Checks Against Overwriting	69
Tape Archives	
Cataloging	42
Tape Data Sets	
Multivolume, Creating	40
Tape Management	
Allocation Considerations	38
Allocation, DDNAMES Used	38
Archive Tapes, Concurrent Duplication	39
Assigning Tape Expiration Dates	43
Cataloging the Tape Archives	42
Creating Multivolume Tape Data Sets	40
Dynamic Allocation Versus Supplied JCL	38
EDRC Support in F6470 Cartridge Devices	40
EDRC, Considerations for Use	41
General Information	37
Running Archive/Backup in Simulate Mode	41
Running Concurrent Archive/Backup Jobs	41
Specifying Blocksizes for Archive/Backup	40
Tape Format	50
Tape GDG Support	42
Tape/Cartridge Switching	48
Tapepool Considerations	43
Tapepool Update Utility	46
Tapepool Updates from ISPF	45
Tape Pool Subfile	528
Tapepool	
Considerations	43
Subfile Record Format	528
Updates from ISPF	45
Tapepool Update Utility	46
ADD Command	47
DELETE Command	47
LIST Command	46
TAPEPULL	571
Threshold	
Calculations, PDS Compression	459
Threshold Manager	90
Commands and Parameters	95
Control Hierarchy	94
Control Statement Syntax	91
Implementation Considerations	99
JCL	95
Overview	90
THRSHMGR	91

User Exit	100
THRSHMGR	91
Track Dump Utility	347
Command and Parameters	348
JCL	347
TSO Dynamic Restore Subfile	528
TSS Command Processor	
TSS/PFD USERID Authorization List	563
Types of Media	
Specifying	60

U

UNLOAD	
Command and Parameters	18
UNLOAD Utility	17
UNLOAD, RELOAD, and REORG Utilities	17
Updates	
Tapepool, from ISPF	45
Tapepool, Utility	46
Used Space Percentage	
Directory	459
Member Data	459
User Exits	213
Archive Fix Utility	342
Archive Tape Analysis Utility	336
Cross-reference	215
Descriptions	221
General Information	213
SYSOUTEX	54
Tailoring for All Sysout Print Lines	54
Threshold Manager	100
User Functions	
Adding to SAMS:Disk PFD	362
User Members, PARMLIB	548
User-specified Condition Codes	88, 564
CCSET	88
SAMPCSET	88
User-written Messages	315
USERID	
TSS/PFD, Authorization List	563
Utilities	
Archive Extract	336
Archive Fix	339
Archive Tape Analysis	333
Archive Tape Recovery	350
FRECOVER	23
Load Library Status	346
Recovery	23

RELOAD	20
REORG	21
Repair Archvols	343
Subfile Dump	27 - 28
Tapepool Update	46
Track Dump	347
UNLOAD	17
UNLOAD, RELOAD, REORG	17
Utility	
Syntax Validation for PARMLIB	547

V

Validation	
System Parameter	360
Volume	
Source, Pool Member, Auto-Restore	552
Specification Table, Retention Control	559
Volume Allocation Summary Report	
Print Table	464
Record Format	463
Volume Attribute Summary Report	
Print Table	470
Record Format	469
Volume Distribution Summary Report	
Print Table	478
Record Format	477
Volume Freespace Report	
Print Table	493
Record Format	492
Volume Layout Report	
Print Table	495
Record Format	494
Volume Mapping Report	
Print Table	497
Record Format	496
VSAM	
Data and Index Component, New Name Algorithm	447
DD Statement	571
Idle Space Calculation	446
VSAMKEY	572
VSAMRVOL	572
VTOC Entries	
and Report Processing	437